Contribution à la modélisation eulérienne unifiée de l’injection : de la zone dense au spray polydispersé

Mohamed Essadki

To cite this version:

Mohamed Essadki. Contribution à la modélisation eulérienne unifiée de l’injection : de la zone dense au spray polydispersé. Mathématiques générales [math.GM]. Université Paris Saclay (COmUE), 2018. Français. NNT : 2018SACLC024 . tel-01928584

HAL Id: tel-01928584
https://theses.hal.science/tel-01928584v1

université
PARIS-SACLAY

CentraleSupélec

NNT : 2018SACLC024

Contribution to a unified Eulerian modeling of fuel injection: from dense liquid to polydisperse evaporating spray

Thèse de doctorat de l'Université Paris-Saclay
préparée à CentraleSupélec

Ecole doctorale n°574 École doctorale de mathématiques Hadamard (EDMH)
Spécialité de doctorat: Mathématiques appliquées

Thèse présentée et soutenue à Gif-sur-Yvette, le 13 Février 2018, par

Mohamed ESSADKI

Composition du Jury :

Frédéric LAGOUTIÈRE Professeur, Université Claude Bernard Lyon 1	President
Daniele MARCHISIO Professeur, Université Politecnico di Torino	Rapporteur
Jean-Marc HERARD Directeur de recherche, EDF-DRD	Rapporteur
Stéphane VINCENT Ingénieur de recherche, Université Paris-Est Marne-la-Vallée	Examinateur
Marc MASSOT Professeur, Ecole Polytechnique	Directeur de thèse
Stephane de CHAISEMARTIN Ingénieur de recherche, IFPEN	Promoteur de thèse
Frédérique LAURENT Directrice de recherche, (CNRS, EM2C)	Co-directrice de thèse
Adam LARAT Ingénieur de recherche, (CNRS, EM2C)	Co-encadrant de thèse
Grégoire ALLAIRE Professeur, Ecole Polytechnique	Invité
François-Xavier DEMOULIN Professeur, Université de Rouen	Invité
Sibendu SOM Ingénieur de recherche, Argonne National Laboratory	Invité

Thèse de doctorat

Remerciements

Je tiens tout d'abord à remercier les membres du jury: D. Marchisio, J.M. Herard, G. Allaire, H. Kuipers, F. Lagoutière, S. Vincent, M. Massot, F. Laurent, S. De Chaisemartin, S. Sibendu et F.X. Demoulin pour l'intérêt qu'ils ont accordé à ce travail. J'ai été très honoré par leur présence et les échanges qu'on a eu à propos de la soutenance et du manuscrit de la thèse.

IFPEN et le laboratoire EM2C à l'école CentraleSupélec, par les moyens humains, matériels et financiers m'ont permis de réaliser cette thèse dans un très bon environnement de travail avec beaucoup de facilités. Je remercie IFPEN de m'avoir financé durant les trois ans de la thèse et de leur accueil dans le département mathématiques appliquées. Je remercie également le laboratoire EM2C, l'école CentraleSupélec et l'école doctorale de mathématiques Hadamard de m'avoir fourni les moyens dont j'ai eu besoin et aussi d'avoir contribué au financement de mes différents déplacements (des formations, des conférences et des séminaires).

Avoir bien réussi une thèse n'aurait pas été possible sans l'équipe d'encadrement. J'ai une grande chance d'avoir des encadrants pas que des bons scientifiques mais aussi des personnes avec des très grandes qualités humaines. Grâce à eux, j'ai pu surmonter les différentes difficultés durant ces trois ans. Je tiens à les remercier très chaleureusement et je resterai toujours reconnaissant à leur aide, leur effort et leur investissement. Je voudrai aussi les remercier de la liberté qu'ils m'ont laissée pour orienter mes travaux de thèse. Merci Marc, Stephane, Frédérique et Adam pour votre confiance, votre écoute et votre attention à la qualité du travail. Je garderai toujours des bons souvenirs de nos échanges, les séances du travail et aussi les moments de soulagement. Je tiens à saluer particulièrement Marc pour sa capacité à gérer les situations de tension et préserver un bon climat d'échange. Ensuite, je remercie Stéphane Jay qui a suivi avec un grand intérêt l'avancement de cette thèse.

Durant les trois années, j'ai eu la chance de travailler avec d'autres personnes que mes encadrants. Tout d'abord, je remercie F. Drui pour sa collaboration. Je tiens aussi à remercier mes collègues du laboratoire de Coria, en particulier F.X Demoulin, T. Ménard, J. Réveillon, B. Duret, S. Puggelli et R. Canu pour les différentes interactions qu'on a eu lors de mes visites à Rouen. Je souhaite aussi remercier J. Jung, A. Larat, M. Pelletier, et V. Perrier avec qui j'ai gardé des bons souvenirs de l'été de CEMRACS 2016.

Je remercie tous mes collègues et les personnels du laboratoire EM2C et IFPEN que j'ai eu le plaisir de côtoyer et d'échanger avec eux durant les trois années de la thèse. Je souhaite une bonne continuation pour tous les doctorants qui ont soutenu leurs thèses et le meilleur pour la fin pour les autres.

Enfin, je remercie ma famille et mes amis de m'avoir soutenu dans mes choix et d'avoir cru en moi. En particulier mes parents que je les remercie de tout mon cœur, merci papa et merci maman.

Résumé : L'injection directe à haute pression du carburant dans les moteurs à combustion interne permet une atomisation compacte et efficace. Dans ce contexte, la simulation numérique de l'injection est devenue un outil fondamental pour la conception industrielle. Cependant, l'écoulement du carburant liquide dans une chambre occupée initialement par l'air est un écoulement diphasique très complexe ; elle implique une très large gamme d'échelles. L'objectif de cette thèse est d'apporter de nouveaux éléments de modélisation et de simulation afin d'envisager une simulation prédictive de ce type d'écoulement avec un coût de calcul abordable dans un contexte industriel. En effet, au vu du coût de calcul prohibitif de la simulation directe de l'ensemble des échelles spatiales et temporelles, nous devons concevoir une gamme de modèles d'ordre réduit prédictifs. En outre, des méthodes numériques robustes, précises et adaptées au calcul de haute performance sont primordiales pour des simulations complexes.

Cette thèse est dédiée au développement d'un modèle d'ordre réduit Eulérien capable de capter tant la polydispersion d'un brouillard de goutte dans la zone dispersée, que la dynamique de l'interface dans le régime de phases séparées. En s'appuyant sur une extension des méthodes de moments d'ordre élevé à des moments fractionnaires qui représentent des quantités géométriques de l'interface, et sur l'utilisation de variables géométriques en sous-échelle dans la zone où l'interface gaz-liquide ne peut plus être complètement résolue, nous proposons une approche unifiée où un ensemble de variables géométriques sont transportées et valides dans les deux régimes d'écoulement. Cette approche permet le traitement d'un brouillard d'objets non sphériques et dégénère naturellement vers un modèle de moment d'ordre élevé fractionnaire dans le cas sphérique ; elle a la même capacité que le modèle EMSM de D. Kah ou le modèle multi-fluide de décrire la polydispersion.

Le développement d'un modèle fiable pour la dynamique de l'interface lorsqu'une échelle de résolution est fixée, repose sur l'analyse de simulation directes issues du code ARCHER du CORIA. Pour cela nous avons conçu un nouvel algorithme d'évaluation des propriétés interfaciales, surface et courbures, à partir d'un champ de fonction distance. Il repose sur la préservation d'invariants géométriques et topologiques de l'interface et permet de passer d'un statistique de l'interface à une statistique d'objets. Il s'agit d'un point clef pour proposer des fermetures de notre modèle réduit.

Ensuite, des schémas numériques d'ordre élevé en espace et en temps, précis et robustes ont été développés. Ils ont la particularité de préserver des espaces convexes et assurent donc la préservation de l'espace des moments d'ordre élevé, propriété que l'on appelle aussi réalisabilité. Cette condition est nécessaire à la reconstruction d'une distribution de taille continue à partir de cet ensemble fini de moments fractionnaire transportés, reconstruction obtenue par maximisation de l'entropie de Shannon. Enfin, nous fournissons un nouvel algorithme pour une résolution précise et robuste de l'évaporation. Un traitement spécifique par rapport au cas des moments entiers est proposé pour assurer la précision. En effet, la dynamique de l'évaporation nécessite de prendre en compte des moments d'ordres négatifs.

Afin de gagner en précision et de réduire le coût du calcul, les modèles développés ont été implémentés dans un code parallèle avec maillage adaptatif dynamique. Les résultats montrent une très bonne performance en calcul parallèle, tout en conservant une résolution précise et robuste. Enfin, une seconde implémentation du modèle dans un programme basé sur des tâches et qui utilise un ordonnanceur d'exécution pour des architecture multi-cœurs hétérogènes, montre que les GPUs peuvent être utilisés pour accélérer les tâches qui nécessitent un calcul à forte intensité arithmétique.

Abstract: Direct fuel injection systems are widely used in combustion engines to better atomize and mix the fuel with the air. The design of new and efficient injectors needs to be assisted with predictive simulations. The fuel injection process involves different two-phase flow regimes that imply a large range of scales. In the context of this PhD, two areas of the flow are formally distinguished: the dense liquid core called separated phases and the polydisperse spray obtained after the atomization. The main challenge consists in simulating the combination of these regimes with an acceptable computational cost. Direct Numerical Simulations, where all the scales need to be solved, lead to a high computational cost for industrial applications. Therefore, modeling is necessary to develop a reduced order model that can describe all regimes of the flow. This also requires major breakthrough in terms of numerical methods and High Performance Computing (HPC).

This PhD investigates Eulerian reduced order models to describe the polydispersion in the disperse phase and the gas-liquid interface in the separated phases. First, we rely on the moment method to model the polydispersion in the downstream region of the flow. Then, we propose a new description of the interface by using geometrical variables. These variables can provide complementary information on the interface geometry with respect to a two-fluid model to simulate the primary atomization. The major contribution of this work consists in using a unified set of variables to describe the two regions: disperse and separated phases. In the case of spherical droplets, we show that this new geometrical approach can degenerate to a moment model similar to Eulerian Multi-Size Model (EMSM). However, the new model involves fractional moments, which require some specific treatments. This model has the same capacity to describe the polydispersion as the previous Eulerian moment models: the EMSM and the multi-fluid model. But, it also enables a geometrical description of the interface.

A novel algorithm to extract some interfacial quantities from a level-set field is developed in this work. The algorithm is consistent with geometrical and topological invariants. It aims at post-processing DNS results of representative configurations. This tool should be helpful in modeling and closing the evolution equations of the interfacial variables.

In terms of numerical methods, the robustness and the accuracy are two important points for a predictive simulation. High order numerical schemes with strong stability properties are proposed in the present work. The proposed methods ensure the preservation of the moment variables in the moments space. This is a necessary condition for the reconstruction of the size distribution from a finite set of its moments. In the present work, we use a continuous reconstruction of the size distribution, which maximizes the Shannon entropy for a given set of fractional moments. Finally, we provide a novel algorithm for an accurate resolution of the evaporation, which requires specific treatment compared to the case of integer moments, since it involves negative order moments.

In order to gain more accuracy and to reduce the computation cost, the fractional moments model, dedicated to the simulation of an evaporating polydisperse spray, is implemented in an Adaptive Mesh Refinement (AMR) and parallel code. The results show a good parallel scalability, while keeping a good resolution using AMR grids. Finally, a second implementation of this model in a task-based program, which uses a runtime scheduler of the tasks in heterogeneous multi-cores, shows that GPUs can accelerate the tasks that require intensive arithmetic computations.

Contents

List of Figures

Chapter 1

Introduction

1.1 General context and main objective

In the last decade, with the large change of the Earth's climate, the increase of energy demand and the growth of the world population, the world faces new challenges to ensure an efficient and rational use of energy. Fossil fuels are the most consumed primary energy sources that have enabled a large industrial and economic growth all over the world. Although different new energy forms have been developed in the recent years to substitute old energy ones, the use of fossil fuels still remains the main energy source for the transport sector. However, the increasing cost of extraction and the depletion of oil wells require new engineering solutions to increase the efficiency of engines and thereby reduce the fuel consumption. Furthermore, the combustion of the fuel implies production of pollutants that are the main cause of the global warming such as CO_2, which is one the most prominent greenhouse gases and nitrogen oxides (NO_x), soot and other particles that are the most relevant for air pollution. In this context, automotive engines are deeply concerned by these issues. Many research and engineering studies are conducted in order to better understand the combustion mechanisms and to develop more efficient engines. One of the most used methods that helped in the past to develop the current technologies consists in experimenting and prototyping new designs, before going toward a large industrial production. While experiments are still a necessary step to develop and improve new technologies, it can imply high costs in the research & development stage. Also, from a feasibility standpoint, taking measurements of an experimental setup can sometimes be very difficult with the current probe technologies. For these reasons, it is relevant to assist the design of new combustion engines with predictive numerical simulations that can bring more information on the combustion regimes as well as on the global behavior of the engine.

The flow in combustion devices involves very complex physical phenomena. Indeed, the multi-scale character is very present at different levels:

- First, the flow is turbulent and it is characterized by high Reynolds numbers $Re_L \sim 10^4$ estimated with respect to the typical dimension of diesel combustion chamber $L \sim 10\text{cm}$. The turbulent scales vary between large scales described by the dimensions of the combustion chamber $L \sim 10\text{cm}$ and the dissipative eddy scales given by the Kolmogorov scale $\eta \sim 20\,\mu\text{m}$.
- The second difficulty involved in the combustion lies in the reactive character of the flow. Combustion is a set of chemical reactions that releases heat energy. Indeed, the very hot

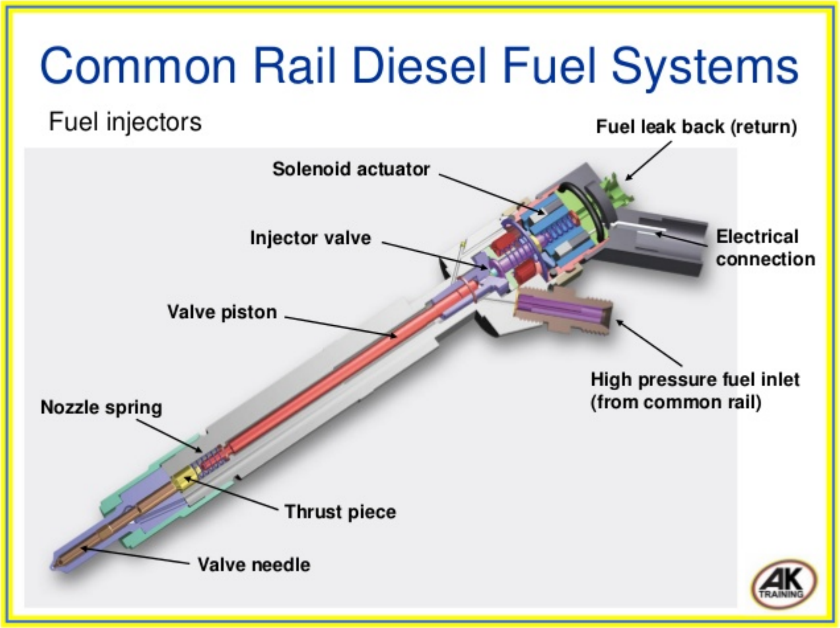


Figure 1.1: *Fuel injector device. Source: "www.slideshare.net/amgadradhihadi/common-rail-diesel-fuel-systems".*

flame in combustion chamber is caused by a highly exothermic reaction taking place in a very thin zone ~ 1mm. The turbulent flow in the combustion chamber and its interaction with an unstable flame front involving a large set of species and reactions generate an even larger spectrum of scales in both space and time.

- Finally, the combustion in direct injection engine involves a complex two-phase flow. The combustion of two phases is significantly different from the purely gaseous one. Indeed, the two-phase flow of the liquid fuel and the air has a direct impact on the combustion regime and on the emission of pollutants. The liquid fuel is injected in the combustion chamber at high velocity. In diesel combustion engine, the two-phase flow regimes are characterized by high Weber numbers[1] $We > 40$ and Ohnesorge numbers[2] $0.04 < Oh < 0.07$. For such configuration, we obtain a wide range of the gas-liquid interface and droplet scales, which are mainly given between the diameter of the fuel injector nozzle ~ 200 µm and the small droplets obtained after the atomization ~ 1 µm.

Even with the high increase of the High Performance Computing (HPC) infrastructures, a Direct Numerical Simulation of such flows, where all the scales need to be solved, is still not affordable in realistic configurations of high Reynolds and Weber numbers. For this reason, the modeling of the turbulence, the combustion and the two-phase flow is necessary to decrease the computational cost. In the present PhD, we focus on the modeling of the multi-scale character of the two-phase flow involved in combustion engines.

Recently, the high pressure direct systems have been widely used to deliver the fuel in the combustion chamber at the right time and with a controlled metering. The main purpose of these devices is to atomize the liquid fuel by generating a spray of small droplets. The spray can be evaporated rapidly compared to a bulk liquid and well mixed with the air. Thereby, it improves

---

[1] Weber number is a dimensionless variable that measures the ratio between disruptive (aerodynamic) and cohesive (surface tension) forces

[2] Ohnesorge number is a dimensionless number that measures the ratio of viscosity forces with surface tension effects.

the combustion efficiency and reduces the emissions of soot and unburnt fuel. Furthermore, modern injection systems use electronic metering to supply the engine by the required amount, depending on the desired power output. An example of modern injector is illustrated in Figure 1.1, where we can distinguish three main injector components:

A. The **nozzle outlet** is a very small hole of a diameter ~100 µm in the injector and is the final element of the injection system before the fuel enters to the combustion chamber.

B. The **valve needle** is slidable part within the injector. In the rest state, the needle tip is loaded onto the nozzle seat by the injector spring combined with hydraulic pressure, which keeps the nozzle orifice closed.

C. The **high pressure fuel pump** is responsible for compressing the fuel to the pressure required for high pressure injection and which can go up to 2000 bar for diesel engine.

The two-phase flow occurring inside the nozzle and the combustion chamber leads to more difficulties in the numerical modeling and simulation of automotive engine flows. First, the fuel flow within the injector is mainly a monophasic liquid phase. But, the rapid opening and closing of the valve and the cross-section variation cause vortices and pressure drop. It results in a local and fast phase transition, known as the cavitation phenomenon. Therefore, the formation of vapor bubbles and pockets was observed inside the carrier liquid phase Sibendu et al.(2011); Le Martelot(2013); Le Martelot et al.(2014). The cavitation is also one of the underlying physics impacting the liquid disintegration at the downstream of the injector. The jet coming out of the nozzle is a bulk liquid which is separated from the gaseous phase inside the combustion chamber and the liquid phase is not immediately atomized. Close to the nozzle outlet, the two-phase flow is called **separated phases**. The atomization starts right at the exit of the flow. Instabilities of different natures contribute more or less to the liquid fragmentation: 1-Kelvin-Helmotz instability due to the difference between the phase velocities, 2-Rayleigh-Taylor instability due to the mass ratio between the two fluids and 3-Plateau-Rayleigh instability due to interface forces that leads to the separation of drops. These instabilities contribute to a nonlinear growth of small gas-liquid interface deformations, which creates unstable ligaments and drops. The atomization in the **separated phases** is called the **primary breakup**. In the transition zone of the fuel injection, we find structures of different scales: bulk liquid, ligaments and drops. As we move downstream of the flow, the bulk liquid disintegrates further to drops and ligaments. The first generated drops are mainly unstable because of their large Weber number. Therefore, these drops can undergo a **secondary breakup**, which yields to a polydisperse spray of droplets (large range of droplet sizes). The typical range of the generated droplet diameters is [1 µm, 50 µm] for diesel engines. In the downstream region, the obtained two-phase flow regime is called **disperse phase** and it consists of spherical and small droplets carried in the continuous gaseous phase. It has been shown that the polydisperse character of the spray has a key influence and should be described in any attempt of modeling such flows. The challenge is to provide a model as well as a numerical strategy, which are able to capture the large scale spectrum (see Figure 1.2) of the separated and disperse phases and to provide predictive numerical simulations. It is as much a scientific challenge as an applicative one.

IFP Energies nouvelles (IFPEN) is widely involved in the development of new predictive models and numerical simulation softwares to contribute and develop solutions for these challenges. The institute has been leading ambitious projects, covering the entire modeling and experiments of automotive engines, going from the interior flow in the nozzle to the complete combustion, power generation and exhaust gas. It has also designed global system simulators to predict the

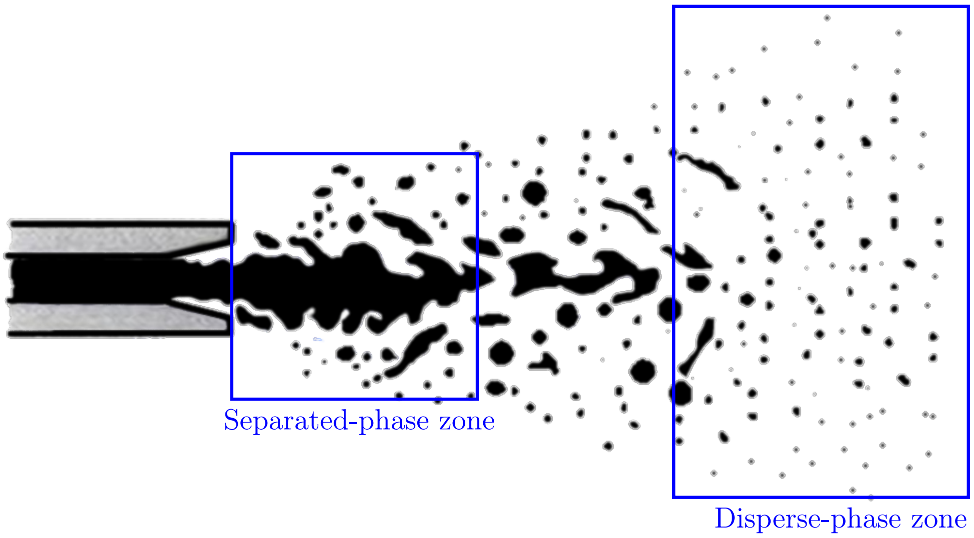


Figure 1.2: *Sketch of a liquid fuel injection and examples of separated-phases and disperse-phase zones, reprinted from Drui(2017).*

global behavior of the system under multiple combustion cycles. IFPEN in collaboration with its institute and laboratory partners conducts different research project to develop innovative solutions for these issues. In the following, we present a short list of these works to emphasize the importance of the fuel injection and two-phase combustion modeling:

- Two-phase flow combustion models were developed for automotive engine applications (diesel and gasoline engines). The ECFM model developed in Colin et al.(2003) for the combustion of perfectly or partially mixed mixtures has shown a great success for gasoline engine simulations, giving the good mixing of the gasoline with the air. In order to extend the model for Diesel applications, the model ECFM3Z Colin and Benkenida(2004); Bohbot et al.(2016) consider three zones: air, fuel and mixing zone to model diffusion flame. The list of IFPEN contributions in this domain is not restricted to these two model examples. But we aim, through this focus, at underlining the importance of the two-phase flows in the combustion modeling, especially for diesel engines where the fuel is not immediately evaporated after the injection.

- The first attempts to use Eulerian models for simulating the full liquid injection were conducted in the industrial code IFP-C3D Vessiller(2007); Truchot(2005) using two-fluid models. The adopted approach was more dedicated to separated phases two-phase flows than to sprays. Even though the method was used to simulate the separated and dispersed phases, the model stays far from an accurate description of the interface topology and the polydisperse character of an evaporating spray. Indeed, the fluid topology is given only through the volume fraction and the surface area density (the expected surface area per unit of volume). Another recent attempt to simulate both the separated and disperse phases, was conducted in Devassy et al.(2015). The authors suggested to consider a two-phase flow model with seven-equation model of Saurel and Abgrall(1999). Then, they couple it with two surface area density equations: one for the disperse phase and the other for the separated phases. However, the strategy to use two different surface area densities depends on the modeling of the exchange terms between the separated phases and the disperse phase. Furthermore, it fails in providing accurate description of the polydispersion.

- For the disperse phase, the most natural way to describe the droplet dynamics is the Lagrangian approach. This approach is widely used to simulate the spray in engines. Lagrangian methods benefit from a simple implementation, do not require sophisticated modeling and do not introduce any numerical diffusion. A Lagrangian description of the spray has been successfully coupled with an Eulerian RANS description of the gas flow in Bohbot et al.(2009); Vié et al.(2010). However, the Lagrangian description is still not appropriate to describe a bulk liquid in the separated phases zone. Recently, new Eulerian models dedicated to polydisperse evaporating sprays de Chaisemartin(2009); Kah(2010); Vié et al.(2013b); Emre(2014) were developed in joint works between the EM2C laboratory and IFPEN. Their approach represents a potential alternative to Lagrangian models that can simulate an evaporating polydisperse spray at reasonable computational cost Massot et al.(2010); Kah et al.(2012). This is a first step to unify the spray description with the separated phases, which is naturally modeled in an Eulerian framework.

This list of contributions shows the involvement of IFPEN and EM2C laboratory in developing numerical models of fuel injection and two-phase flow combustion. We would like to clarify that by this list, we mainly focused on IFPEN and EM2C laboratory projects to present the general context of this PhD, while a more general state of the art will be presented later. This brief review points out the fact that mainly each type of model is suited for one flow area: separated and disperse phase. And even with some recent contributions to couple the models, we are still far from having reached a unified model that can simulate both regions simultaneously. It has indeed been a challenge to correctly model the atomization and the polydispersion in an industrial context with reasonable computational resources. In this PhD thesis, conducted jointly at IFPEN, EM2C laboratory and CMAP laboratory, we aim at contributing and bringing innovative solutions to develop a new numerical model that can tackle simultaneously the different injection zones and can be implemented in High Performance Computing (HPC) applications. Such a model needs to consider the main physical phenomena that control the main flow characters: the interaction between the gas and the liquid, the interface topology evolution, the polydisperse character of the disperse phase and the related phenomena as evaporation, heating and drag force. Furthermore, we aim at ensuring the well-posedness of the model and developing robust and accurate numerical schemes. Before detailing this PhD contribution, we first present in the following section a general state of the art of different contributions in numerical modeling of two-phase flows. In this review, we will particularly focus on the modeling of the two regimes: separated phases and disperse phase.

1.2 State of the art of numerical modeling of two-phase flows

A possible classification of two-phase flow models can be conducted through a separation of two-phase flow regimes: **separated phases** or **disperse liquid phase**. In the literature, we find different types of two-phase flow models. Each approach depends on the flow regime as well as the physical phenomena that need to be correctly captured. We first classify the models in two categories:

- DNS two-phase flow models: in this category of model, we need to solve all the scales. We underline that the smallest scale here is not only defined by the smallest eddy as for turbulent monophasic flows, but also depends on the smallest droplet diameter.
- Reduced order models: we mean by this class all models that use macroscopic quantities

to describe the flow without the need to solve all the scales. In this case, we often resolve large scales whereas below a give spatial scale, the sub-scales phenomena are modeled.

In the following, we discuss in more details these different methods. Along this review, we emphasize the industrial and academic context of the model applications as well as the use of the different approaches to simulate the separated or/and disperse phases.

1.2.1 DNS two-phase flow models

In Direct Numerical Simulation (DNS) of two-phase flows, each phase dynamics is resolved separately through monophasic Navier-Stokes equations, while the jumps in the property fields are properly handled across the interface. The jump relations between the two phases should include surface tension effects. In this class of methods, the gas-liquid interface has to be determined with appropriate techniques. These methods are thus well-suited for separated-phase configurations. When it comes to moving boundary problems, one typically distinguishes between two approaches: **interface tracking** and **interface capturing**.

- **Interface tracking** methods: these methods treat the interface as a sharp interface whose motion is followed explicitly either by a moving grid that follows the fluid motion or by using Lagrangian-markers at the interface. We refer here to the most common tracking methods: Front Tracking method Unverdi and Tryggvason(1992); Hirt et al.(1974); Pianet et al.(2010) and Marker-and-cell (MAC) scheme Harlow and Welch(1965). These methods do not lead to diffuse the interface, but can require sophisticated geometrical algorithms to handle topology change of the gas-liquid interface (interface pinching or merging, breakup and coalescence).
- **Interface capturing** methods: in this type of methods, the interface is not determined explicitly, but instead it is represented thanks to a scalar function. The most common methods are VOF method Hirt and Nichols(1981); Agbaglah et al.(2017), Level Set method Ghods and Herrmann(2013) or a combined VOF and Level Set method Menard et al.(2007); Lebas et al.(2009). In VOF method, the volume fraction of the cell occupied by one of the phases can be calculated by solving a transport equation. The transport equation of the volume fraction is often derived from the mass conservation equation by considering incompressible or weakly compressible liquid phase. VOF method is a mass conservative method but it can suffer from high numerical diffusion. Level Set method uses a signed distance function to the interface, such that it gives a signed distance between each point in the domain and the gas-liquid interface. For example a positive distance in the liquid sub-domain and a negative distance in the gas sub-domain. The distance function is advected with the fluid velocity. At the interface, the Level-Set function is equal to zero and the fluid velocity defines a gas-liquid interface velocity in the case when we do not consider phase transition. However, to ensure that the function remains a signed distance function a redistancing algorithm is applied. Compared to VOF method, Level-Set is more accurate in capturing the gas-liquid interface but it can suffer from mass loss. For this reason, a Combined Level Set and VOF method can be used to gain accuracy and to ensure mass conservation of primary breakup simulations (see Figure 1.3). Interface capturing methods can handle efficiently topological changes without the need for additional treatments. However the breakup process is mesh depending and the mesh convergence is still an open issue for this type of methods.

DNS two-phase flow codes, such as the ARCHER code Menard et al.(2007); Vaudor et al.(2017), may be used for simulations of some academic simple injection configurations. The results of

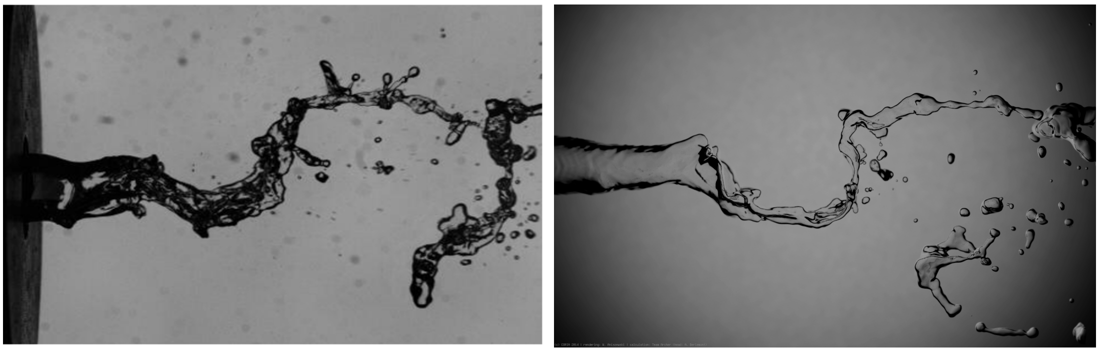


Figure 1.3: *Comparison between experimental results Matas and Cartellier(2013) (left) and DNS simulation of a liquid jet atomization through the hybrid VOF/Level Set sharp interface approach Vaudor et al.(2017) (right).*

these simulations help to better understand the phenomena, since accurate information on detailed physics is sometimes hard to extract from experimental measurements Lebas et al.(2009); Fuster et al.(2009); Desjardins et al.(2013); Ghods and Herrmann(2013); Le Chenadec and Pitsch(2013); Vaudor et al.(2017). However, configurations at large Reynolds and Weber numbers, that characterize a turbulent flow and high interface instabilities, may be extremely costly to compute. Besides, it may fail in predicting the smallest interfacial structures, such as the very small droplets or thin ligaments. Therefore, while DNS is of great interest in academic research, it is not appropriate for a direct industrial use.

1.2.2 Reduced order models

Reduced-order models intend to avoid the simulation of the smallest scales of the configuration by providing a description and the evolution laws only for some macroscopic quantities of interest. For example, the volume fraction of the phases in mixture zones and the surface area density of the two-phase interface can be used to describe the gas-liquid interface. So far, however, these models inherently depend on the two-phase flow regime: **separated** or **disperse** phase regimes. Building up a multi-scale and accurate model with the capacity of resolving the whole injection process is a challenging task, that can be addressed either by coupling models associated to the two main flow classes, namely the disperse and separated phase flows, or by developing a unified approach.

1.2.2.1 Reduced order models for the disperse phase

The main challenge in simulating the disperse phase consists in describing some droplet relevant properties as the size (polydispersion), the velocity (polykinetic) and the temperature distributions as well as the gas-droplet interactions, while ensuring a reasonable computational cost. In reduced order models of the disperse phase, the gas-liquid interface and the flow surrounding the droplets are not resolved. Furthermore, we often suppose a spherical shape of droplets. Two levels of modeling can be envisioned, based on a deterministic or a probabilistic approach.

- For a deterministic approach, the droplets are tracked in the flow using a pure Lagrangian method. Each droplet is tracked by solving the evolution of its center position, velocity, size, temperature, etc. We refer to this method by the Discrete Particle Simulation (DPS) Mashayek(1998); Zhu et al.(2007); García et al.(2007); Zamansky et al.(2016). Deterministic Lagrangian methods combine an efficient modeling of the polydispersion and the polykinetic features of the flow and high numerical resolution since they do not introduce numerical diffusion as in the case of Eulerian methods. However, Lagrangian methods suffer from important drawbacks: 1- the coupling with Eulerian description of the gas is still an open question since it involves two ways of description that are fundamentally different, 2- the method can require high computational cost when we use a large number of droplets and 3- complex and costly dynamic load balancing algorithms are needed to ensure a good parallel computation scaling.

- On the other hand, probabilistic approaches, also called kinetic-based models, can be used to simulate the disperse phase flow. They rely on a number density function (NDF), that satisfies a generalized population balance equation (GPBE), also known as the Williams-Boltzmann Equation (WBE) Williams(1958). The phase space of the NDF can include different physical properties of the disperse droplets such as velocity, size, temperature, etc. However, the large phase-space dimension makes the direct resolution by deterministic methods of the WBE quickly unaffordable. Within this category, a wide range of methods have been used to reduce the dimension and resolve the WBE with a reasonable computational cost. First, the stochastic Lagrangian Monte-Carlo approach Bird(1994) is based on samples of representative particles, which are tracked using a Lagrangian method and that allows to estimate the evolution of NDF. Even though this approach can be considered as the most accurate for solving WBE, it still suffers from the same drawbacks as the deterministic Lagrangian methods. In order to cope with these difficulties, one can use a Eulerian kinetic-based model. Two main features need to be correctly modeled in a Eulerian framework: the polydispersion and the polykinetic features. In this review, we restrict the discussion to the modeling contributions of the polydispersion, since it is a key feature for spray combustion models Vié et al.(2013a); Hannebique et al.(2013) and the reader can refer to chapter 3 for more details on the polykinetic modeling. In Eulerian modeling, the polydispersion is described using macroscopic/statistical quantities. The main difficulty consists in choosing the relevant information to capture the size distribution. In the literature, three type of Eulerian kinetic-based models for the description of spray polydispersity can be found:

  A. sectional or multi-fluid models consist in discretizing the size phase space into size bins, called *sections* since the work of Greenberg et al.(1993). In this class of methods, WBE is integrated over each section to derive equations on moments up to the second order defined for each section de Chaisemartin et al.(2009); Doisneau(2013); Laurent et al.(2016); Sibra(2015).

  B. quadrature-moment methods such as QMOM, CQMOM or DQMOM Marchisio and Fox(2005) consider the NDF as a sum of Dirac-delta functions,

  C. EQMOM Nguyen et al.(2016); Yuan et al.(2012) propose a continuous reconstruction of the NDF by extending the Dirac-delta functions to kernels,

  D. high order moments with a continuous reconstruction of the size distribution developed in Kah et al.(2012); Emre et al.(2015); Massot et al.(2010); Vié et al.(2013b). At each step, a continuous NDF maximizing the Shannon entropy is reconstructed from the high order moments Mead and Papanicolaou(1984), with a complete coverage of the whole moment space.

These different methods rely on statistical information of the size distribution given by its moments, instead of solving directly the WBE. In this PhD, we choose to use the high order moments with a continuous reconstruction through entropy maximization. In fact, this method avoids to use several sections and saves computational time Kah(2010); Kah et al.(2015). It is also possible to use a hybrid method by using few sections to capture the size-velocity correlation (one velocity by size-section). Moreover, the continuous reconstruction allows a more accurate and consistent evaluation of the evaporation flux, compared to the discontinuous approach used in the quadrature-moment methods, as well as a comprehensive description of the moment space. This method will be discussed further in chapter 3 and compared with the two other Eulerian methods.

1.2.2.2 Reduced order models for separated phases

Among the various approaches that may be used in separated-phases regimes, where the dynamics of an interface has to be resolved, or at least its main features, let us mention the **two-fluid models** or the **interface mixture models**. The two-fluid models, usually denoted 6- or 7-equation models refer to averaged two-phase models that consider two velocities: one velocity by phase, while interface mixture models suppose an equilibrium between the two velocities and can be obtained through a relaxation process from the 6- or 7-equations models. In the present thesis, we will simply denote the two classes of models "two-fluid models".

Such two-fluid models are given by various systems of PDEs that describe the evolution of averaged quantities of the flow as in Chanteperdrix et al.(2002); Murrone and Guillard(2005); Bernard-Champmartin and De Vuyst(2014). These equations may be derived through an averaging procedure Ishii(1975); Drew and Passman(1999), or by using a variational principle Gavrilyuk and Saurel(2002); Drui et al.(2016b). In both cases, they stand for a spatial-, temporal- or ensemble-averaged two-phase flow. This class of models describe a mixture of the two phases and can not in general provide a sharp representation of the interface, which is smoothed out by the averaging process. Indeed, the two phases can be present at any given location, according to the values of a characteristic function, that is generally the volume fraction of one of the phases. Traditionally, these models provide little information about the sub-scale interfacial structures. The volume fraction is often the only variable used to describe the flow topology.

Recent works, as in Drui et al.(2016b), have shown that they can be enriched with further sub-scale physics and can describe some disperse phase regimes. Yet, such models are far from the ability to deal with polydispersity, in the way kinetic-based models and particularly Eulerian moment models do. From a numerical point of view, two-fluid models are used for simulations of separated phases and interfacial configurations, where the exact location of the interface is not reconstructed, but lies in a mixture zone due to the modeling and numerical approximation. Although the numerical diffusion can be reduced by using more accurate numerical schemes, the spreading of the interface is still a main bottleneck of these methods for the simulation of atomization.

There also exists some compressible models, which can deal with "sharp" interface description and for which the volume fraction is either 0 or 1 such as in Allaire et al.(2002); Kokh and Lagoutière(2010), as well as Chanteperdrix et al.(2002); Drui et al.(2016b). Even is such models yield some maximum principles and admit some well-posed discontinuous solutions in volume fraction, they can not really constitute a DNS-like model for our purpose and some of these models can also be interpreted as mixture models such as in Chanteperdrix et al.(2002); Drui et al.(2016b).

1.2.2.3 Some contributions to couple separated phases and disperse phase approaches

Recent works have been devoted to the numerical coupling between separated phases and disperse phase approaches. Among them, Le Touze et al Le Touze(2015) proposed to couple a two-fluid model for separated phases with a multi-fluid model for the disperse phase. Up to now, the exchange terms between both models depend on the configuration of the atomization and cannot predict the generated distribution in size of the disperse droplets from the atomization. One can also mention the Eulerian-Lagrangian Spray Atomization (ELSA) technique Vallet and Borghi(1999); Vallet et al.(2001); Lebas et al.(2009), where a two-fluid model is enhanced with an equation for the expected surface area density in the dense zones. This model is then coupled with a Lagrangian approach for the simulation of the disperse phase in the dilute zones. In Devassy et al.(2015), the same set of equations is used with a differentiation between the variables describing the disperse and separated phases. Two additional equations on the expected density area are also used: one for the separated phases and one for the disperse phase. So far, these approaches do not provide a unified description of the whole atomization process (from the separated phase to the spray of droplets) and fail in providing an accurate description of the polydispersion for the disperse phase. Indeed, in the works presented above, the description of the gas-liquid interface geometry relies on one or two variables only, that are the volume fraction and the expected surface area density. This information is not sufficient to reconstruct a NDF of a polydisperse spray.

1.2.3 Adaptive Mesh Refinement techniques applied for two-phase flow simulations

The accuracy of the numerical resolution is an important point for predictive simulations and especially for Eulerian two-phase flow simulations. In the separated phases, it is important to capture accurately the gas-liquid interface dynamics and to limit the numerical diffusion of the interface. While for the disperse phase, the concentration of droplets is a key feature for predictive combustion simulations. One of the possible solutions consists in using a high order numerical method to limit the numerical diffusion. Nevertheless the spatial representation of the flow using sufficient spatial discretization plays a crucial role to properly describe the physical dynamics of the problem. Furthermore, the numerical diffusion of the interface can not be completely eliminated using only high order numerical scheme. Recently, Adaptive Mesh Refinement (AMR) techniques have been widely used to simulate two-phase flows in very complex simulations that require a high numerical resolution Drui(2017); Zuzio and Estivalezes(2011); Herrmann(2010); Herrmann(2008). These techniques allow an accurate spatial representation of the flow where it is needed while saving computational time and memory. However, handling dynamically adapting grids increases the complexity compared to fixed ones. The adaptation of the meshes leads to high overheads (the excess time spent in AMR specific use in comparison with an equivalent fixed grid use). Furthermore, a good load balancing algorithm is required to ensure a high parallel scalability. Let us just mention here that recent developments of AMR codes and libraries have lead to highly scalable methods, well-suited for HPC simulations. For example the `p4est` library Burstedde et al.(2011) was shown to scale up to 458,000 cores. We present in more details this library and the different challenges facing AMR applications in chapter 9.

1.3 Main contributions and manuscript organization

According to the above discussion and the comparison between DNS and reduced order models or between Lagrangian and Eulerian models, we make the choice of developing **Eulerian reduced order** models. Indeed, Eulerian framework is more suitable to design a unified model for the disperse and separated phases. Furthermore, in industrial applications, DNS models lead to high computational cost, which we may reduce thanks to Eulerian reduced order models. Nevertheless, the discussion has also highlighted two majors difficulties encountered in Eulerian reduced order two-phase flow modeling: the first one concerns the need of efficient modeling of the spray polydispersion in the disperse phase, while the second concerns the description of the gas-liquid interface topology in the separated phases region that can allow an efficient modeling of the atomization. For the polydispersion, we rely on the pioneering work of Kah(2010), where a set of high order size-moments with a continuous reconstruction of the size distribution is used to model an evaporating polydisperse spray. However, this approach is restricted to spherical droplets. Therefore, an extension to a more general gas-liquid interface topology would allow an efficient coupling with a separated phases model and also enhance the sub-scale description of the gas-liquid interface.

The contributions of this PhD in the numerical modeling and the simulation of the whole injection process cover three different fields: mathematical modeling, numerical methods and high performance computing. These contributions are detailed in four parts as follows:

- **Two phase flows modeling**: In this first part, we provide a detailed analysis of the existing two fluid and moment-method models, to simulate respectively separated and disperse phases. The objective of this first study is to determine the key features of the two regions and to investigate the main difficulties encountered for the two models. The review will allow to identify the difficulties in coupling the two approaches and to choose a relevant strategy that aims at describing both regions with a similar set of variables. First, chapter 2 provides a general derivation of two-fluid models based on the ensemble averaging process. We discuss the main mathematical and physical properties of the obtained models. In this chapter, we also highlight the main difficulties in capturing the interface because of the numerical diffusion. Special treatments are proposed in the literature to limit the numerical diffusion of the interface. Although, in two-fluid models, the interface described by the volume fraction will spread due the large deformations and instabilities of the interface. In this case, some contribution propose to use additional variables as the surface area density to enhance the gas-liquid interface description. However, in chapter 3, we show that these two variables are not sufficient to describe the polydispersion in the disperse phase. The kinetic-based approach of spray and the derivation of moment models from the Williams-Boltzmann equation is detailed in this chapter. We present the different moment models that are used in the literature to tackle some common and known issues related to Eulerian spray modeling such as the polydispersion and the polykinetic features. The main conclusion to be drawn from this study is that: we need to come up with a unified set of macroscopic quantities to describe the flow in separated and disperse phases, such that the variables can provide a sufficient description of the polydispersion and enrich the interface description of two-fluid models.
- **Contribution to a unified modeling of disperse and separated phases**: In the second part, we propose to study the possibility of describing gas-liquid interfaces by using new geometrical information, such as the volume fraction, surface area density and curvatures (mean and Gauss curvatures). In chapter 4, we show that these geometrical variables can be related to high-order fractional moments for spray of spherical droplets.

We also introduce the mathematical properties of the model and show that we can preserve all the advantages of the previously introduced high order moment methods Kah(2010). Moreover, we generalize useful properties of the fractional moment space such as canonical moments as well as lower principal representation Dette and Studden(1997). These properties are relevant ingredients to design high order robust schemes and algorithms that solve a high order moment system. Finally, we extend the continuous reconstruction of the NDF through the maximization of Shannon entropy to this new set of moments. Inspired by the pioneering works of Pope(1988) for the description of the dynamics of flames, in chapter 5, we define a Surface Density Function (SDF) within a different phase space: in our work, it is composed of the mean and Gauss curvatures and the interfacial velocity. The key issue is to make the link between the statistics of a local description of the interface through geometrical variables, such as curvatures, and the statistical description of isolated objects through a number density function in an appropriate phase space. Using Gauss-Bonnet formula, we show that such link is possible between the two distributions. The link between the two different statistics proposes new perspectives and options to develop a unified approach for the two regimes of the two-phase flows. We also propose, in the same chapter, a new numerical procedure for the computation of the curvatures and of the different statistical distributions (SDF or NDF, based on a characteristic spatial averaging size) from the values of a Level Set function. These new algorithms, which propose an evaluation of curvatures and surface area coherent with geometrical and topological invariant, are eventually applied to the post-processing of some DNS simulations, obtained with the ARCHER code Menard et al.(2007); Vaudor et al.(2017). Finally, in chapter 6, we present some work in progress in order to develop a unified model for both disperse and separated phases. The objective of these contributions is to develop a sub-scale model for the interface based on geometrical moments proposed in chapter 5 and coupling this description with a two-fluid model, which degenerates toward a "sharp" interface model, when no sub-scale model is needed. We also emphasize the importance to extract the curvatures and their evolutions from DNS simulations, in order to propose closure relations of this sub-scale model.

- **Numerical methods**: In this part, we focus on the numerical resolution of the fractional moment model to simulate a polydisperse evaporating spray of spherical droplets. For this purpose, splitting techniques are used to split the resolution of the convective part and the source terms. In chapter 7, we propose two different scheme types to numerically resolve the convective part of the system of equations: Kinetic Finite Volume (KFV) scheme and Runge-Kutta Discontinuous Galerkin (RKDG) scheme. The KFV scheme is based on a kinetic approach Bouchut et al.(2003); de Chaisemartin(2009) to derive expressions of the numerical fluxes used on the finite volume scheme. We rely on the work of Kah et al.(2012) to design numerical scheme for the transport equations of the moments up to the second order accuracy, while ensuring the realizability of the moments: each component of the moment vector is a moment of the same positive NDF. We also show how to go further in the accuracy order by using RKDG scheme, while ensuring the realizability of the scheme. For the two schemes (KFV and RKDG), we succeed in ensuring the realizability of the moments and the robustness of the schemes in critical situation as Delta-shock due to a monokinetic assumption. We show these results in a set of 1D test cases, where a detailed study of the accuracy order is discussed. Chapter 8 presents the design of original numerical tools that achieve a very accurate resolution of the evaporation and the drag force, while ensuring the realizability of the moments. A new realizable algorithm to solve the evolution of fractional moments due to the evaporation is proposed. It involves negative order moments and requires an original strategy compared to the integer moment problem Massot et al.(2010). The accuracy and robustness of the proposed strategy is then assessed by a careful investigation of the numerical errors as well as a detailed comparison with the original approach in 0D and 2D academic configurations.

- **High performance computing and adaptive mesh refinement**: In the last part, we present the implementation of the fractional moments in two parallel academic codes that aim at saving computational resources and speedup the numerical resolution. In chapter 9, we present `CanoP` code Drui(2017). It is a C/C++ code, developed within the collaboration of Maison de la Simulation, IFPEN, EM2C and CMAP laboratories. It is based on `p4est` Burstedde et al.(2011), a library providing Adaptive Mesh Refinement (AMR) capability highly scalable in massively parallel computations. The development of `CanoP` was achieved with the idea of a generic framework in which all future models and schemes could be integrated. The `CanoP` code and its first performance results are presented for the fractional moment model to simulate an evaporating polydisperse spray, while further performance results using a two-fluid model can be found in Drui(2017). Finally in chapter 10, we present a task-based program implementation in a 2D code using StarPU library Augonnet et al.(2011) to schedule the tasks among CPUs and GPUs. The main result of this second work underlines the capacity of the GPUs to speedup some tasks that require intensive arithmetic computation as the resolution of the evaporation of the spray model and the benefit of using Runtime tasks Scheduler to optimize the computational time.

This present work was supported by a grant from IFP Energies nouvelles, and benefited from a support from EM2C laboratory, Ecole de mathématiques d'HADAMARD (EDMH) and ANR (Modélisation de l'atomisation d'un jet liquide avec transition sous- et super-critique "ANR-14-CE22-0014", project coordinator: Dr. Thomas Schmitt). The support of EDMH for CEMER-ACS 2016 is gratefully acknowledge.

These contributions resulted in publications in international journals and proceedings:

- *M. Essadki, S. de Chaisemartin, F. Laurent, A. Larat, M. Massot, and S. Jay. High order moment methods and adaptive mesh refinement for polydisperse evaporating spray simulations. Oil & Gas Science and Technology 71 (5) , 2016.*
- *M. Essadki, S. de Chaisemartin, F. Laurent, and M. Massot. High order moment model for polydisperse evaporating sprays towards interfacial geometry description. in revision for SIAM Journal on Applied Mathematics, 2017. Available in "https://arxiv.org/abs/1608.07148".*
- *M. Essadki, F. Drui, S. de Chaisemartin, A. Larat, T. Ménard, and M. Massot. Statistical modeling of the gas-liquid interface using geometrical variables: toward a unified description of the disperse and separated phase flows, submitted to IJMF. Submitted to International Journal of Multiphase-Flow, 2017. Available in "https://arxiv.org/abs/1710.04585".*
- *M. Essadki, J. Jung, A. Larat, M. Pelletier and V. Perrier. A Task-driven implementation of a numerical solver for hyperbolic conservation laws. ESAIM Proceedings 2017, pp. 1-16*
- *R. Canu, C. Dumouchel, B. Duret, M. Essadki, M. Massot, T. Ménard, S. Puggelli, J. Reveillon, and F. Demoulin. Where does the drop size distribution come from? ILASS Conference paper, pages 1–8, 2017*

Part I

Two phase flows models

Chapter 2

Two-fluid models for separated phases

2.1 Introduction

In this chapter, we present a specific class of two-fluid models to simulate the separated phases regime. At a macroscopic scale, the real gas-liquid interface is a sharp surface. However, the impact of different sources of fluctuations and uncertainties makes its exact location hard to predict. In this case, the interface in two fluid models is represented by a thick layer. Indeed, the two-fluid models can be interpreted by considering a statistical averaging across multiple possible realizations of the flow. Each realization can have a different position of the gas-liquid interface depending on the fluctuations and the interface instabilities. In this representation, we can consider a spreading interface that represents the average of different sharp interface realizations.

The objective of this chapter is first to provide a general derivation of two-fluid models based on the ensemble averaging process. We highlight some important properties that a physically and mathematically consistent model should satisfy. In the present work, we consider hyperbolic models for which we can derive an entropy equation. A model that satisfies these requirements is most likely to be a well-posed model. On the other hand, the closure of the source terms should be physically consistent with the second principle of thermodynamics. Finally, our modeling strategy consists in keeping the interface as sharp as possible, when the flow uncertainties (initial conditions, geometry, instabilities, turbulence, etc.) are small enough. This is the case of the gas-liquid interface that is very close to the nozzle. For this reason, it is important to counter the artificial diffusion due to the numerical resolution. Special treatments are proposed in the literature Kokh and Lagoutière(2010); Shukla et al.(2010); So et al.(2012); Tiwari et al.(2013)[1] to sharpen the interface and limit the numerical diffusion. But as we move downstream, the interface instabilities increase and lead to high fluctuations and interface instabilities. In this zone, the two-fluid models diffuse the volume fraction of one of the two phases. This variable represents a statistical averaging over multiple realizations of the presence function of this phase. To enhance this statistical representation of the interface, additional interfacial variables as the surface area density can be used to better account for the interface topology.

---

[1] The methods cited here are different and applied for different class of models that are not necessary suitable for two-fluid models.

2.2 Local instantaneous formulation of each phase

In this section, we consider two separated phases occupying the domain Ω. Each phase $k = l, g$, where l stands for the liquid phase and g for the gaseous phase, occupies a sub-domain $\Omega_k(t) \subset \Omega$, where t is the time. We assume a zero thickness of the gas-liquid interface. Consequently, the gas-liquid interface $\partial\Omega_k = \left(\Omega_l \cap \Omega_g\right) \setminus \partial\Omega$ defines a surface embedded in a 3D domain and we have:

$$\Omega_l \cup \Omega_g = \Omega, \tag{2.1}$$

this equation expresses the saturation of the domain by the two phases.

2.2.1 Local conservative equations

The conservative equations derived in each sub-domain Ω_k are the classical single phase Navier-Stokes equations. The mass, the momentum and the energy equations can be given in the following form:

$$\partial_t \rho_k \phi + \boldsymbol{\nabla_x} \cdot (\rho_k \phi \boldsymbol{u}_k) = \boldsymbol{\nabla_x} \cdot (\psi) + S, \tag{2.2}$$

where ρ_k is the mass density and $\boldsymbol{u}_k$ is the velocity of the phase k. The variables ϕ, ψ and S are summarized in the following table:

Equation	ϕ	ψ	S
Mass	1	0	0
Momentum	$\boldsymbol{u}_k$	$\boldsymbol{T}_k = -p_k \boldsymbol{I} + \boldsymbol{\tau}_k$	$\rho_k \boldsymbol{g}$
Energy	$e_k + 1/2\boldsymbol{u}_k^2$	$-\boldsymbol{T}_k \cdot \boldsymbol{u}_k + \boldsymbol{q}$	$\rho_k \boldsymbol{g} \cdot \boldsymbol{u}_k + r$

where p_k is the pressure of the phase k, $\boldsymbol{g}$ the body force per unit mass, $\boldsymbol{\tau}_k$ the viscous stress tensor, r an additional volumetric energy source term and $\boldsymbol{q}$ the conductivity thermal flux. This set of equations is valid for the phase k, when $\boldsymbol{x} \in \Omega_k(t)$. Closure relations are needed to fully close the system of equations. Considering compressible flows, equation of states for each phase are used to relate the pressure to the mass density or/and the internal energy. Other closure relations are related to the physical fluid properties. For example, the viscous stress tensor can be related to the velocity derivatives in the case of Newtonian-fluid Candel(1990).

$$\boldsymbol{\tau}_k = \mu_k \left(\boldsymbol{\nabla_x}(\boldsymbol{u}_k) + \boldsymbol{\nabla_x}(\boldsymbol{u}_k)^t\right) - \frac{2}{3}\mu_k \boldsymbol{\nabla_x} \cdot (\boldsymbol{u}_k)\boldsymbol{I}, \tag{2.3}$$

where μ_k is the dynamic viscosity of the phase k.

2.2.2 Gas-liquid interface and jump conditions

The gas liquid interface is represented by a surface that separates the two fluids. A deterministic characterization of the interface is given by the characteristic function:

$$\chi_k(t, \boldsymbol{x}) = \begin{cases} 1, & \text{if } \boldsymbol{x} \in \Omega_k \\ 0, & \text{otherwise} \end{cases} \tag{2.4}$$

The characteristic function satisfies the following differential equation, in the sense of distributions Drew and Passman(1999):

$$\partial_t \chi_k + \mathbf{v}_I \cdot \nabla_{\boldsymbol{x}}(\chi_k) = 0, \tag{2.5}$$

where $\mathbf{v}_I$ is the interface velocity.

Boundary conditions are needed at the frontiers of each sub-domain Ω_k. These conditions are written as jump relations across the interface. A derivation of jump conditions can be found in Drew and Passman(1999) (chapter 2 and 8). Here, we express these relations in the case where the phase transition is neglected:

A. jump condition for mass:

$$[\rho(\boldsymbol{u} - \mathbf{v}_I) \cdot \boldsymbol{n}]_I = 0, \tag{2.6}$$

B. jump condition for momentum:

$$[\rho \boldsymbol{u}(\boldsymbol{u} - \mathbf{v}_I) \cdot \boldsymbol{n} - \boldsymbol{T} \cdot \boldsymbol{n}]_I = \sigma H \boldsymbol{n}, \tag{2.7}$$

C. jump condition for energy:

$$[\rho \left(e + 1/2\boldsymbol{u}^2\right) (\boldsymbol{u} - \mathbf{v}_I) \cdot \boldsymbol{n} - (\boldsymbol{T} \cdot \boldsymbol{u})\boldsymbol{n}]_I = \sigma H \boldsymbol{n} \cdot \mathbf{v}_I + \nabla_{\boldsymbol{x}} \cdot (\sigma \mathbf{v}_I) - \frac{d_I e_i}{dt} - e_i \nabla_s \cdot (\mathbf{v}_I), \tag{2.8}$$

where $[]_I$ denotes the jump condition across the interface, e_i the surface internal energy density, σ the surface tension, H the mean curvature of the interface, $\boldsymbol{n}$ the normal at the interface, $\frac{d_I \bullet}{dt} = \frac{\partial \bullet}{\partial t} + \mathbf{v}_I \cdot \nabla_{\boldsymbol{x}} \cdot (\bullet)$ the material derivative at the interface and $\nabla_s \cdot (\bullet)$ the tangential divergence at its surface.

In the context of DNS Menard et al.(2007); Pianet et al.(2010); Agbaglah et al.(2017), the interface can be determined explicitly using some specific tracking or capturing interface techniques. One can then directly solve the local instantaneous equation (2.2) and use jump relations (2.6)-(2.8) to couple the two phases. However, these methods are very expensive in terms of computational cost and provide details which are not required in an industrial context. For these reasons, we focus on the derivation and the modeling of averaged transport equations, where not all the scales are resolved.

2.3 Averaging conservative equations

2.3.1 Ensemble average

Due to many uncertainties (initial conditions, boundary conditions, fluctuations of some source terms and turbulence), a two-phase flow experiment can have many possible realizations, which differ essentially at small scales. These differences can be observed at microscopic scales (very small scales compared to the apparatus scale), even for two exactly similar experimental setups. However, the measurements at large scales are reproducible. Often, for engineering interests only averaged/macroscopic quantities matter. The solutions are indistinguishable at this scale. Hence, it is convenient to consider an averaging representation of the flow (volume, time or statistical averaging). In the following, we introduce the concept of the ensemble averaging, which is a statistical averaging approach. The result of one experiment is called a *realization*. The space of all realizations is parametrized by a certain variable μ. The ensemble of experiments of the same system setup, where the large and medium scales are very similar and indistinguishable from a macroscopic point of view, is called the *realization ensemble* and denoted by $\mathcal{E}$. For a given realization $\mu \in \mathcal{E}$, a field variable of the flow $f(t, \boldsymbol{x}; \mu)$ depends on the realization μ. Finally, we suppose that the realization ensemble is measurable (see Drew and Passman(1999) for more details) and we can evaluate the probability of any realization event $A \subset \mathcal{E}$, as follows

$$P(\mu \in A) = \int_{\mu \in A} dm(\mu), \tag{2.9}$$

where m is the associated probability measure which satisfies $\int_{\mu \in \mathcal{E}} dm(\mu) = 1$.

Based on these definitions, we define the *ensemble averaging operator* $\langle \bullet \rangle$. We apply this operator to the field on realizations $f(t, \boldsymbol{x}; \mu)$ to obtain the averaged / macroscopic field:

$$\langle f \rangle (t, \boldsymbol{x}) = \int_{\mu \in \mathcal{E}} f(t, \boldsymbol{x}; \mu) dm(\mu). \tag{2.10}$$

In practice, we only have a finite ensemble of experiments or simulations. Therefore, we approximate the ensemble averaging operator through a simple averaging over a finite sample of realizations $\mathcal{E}_d = (\mu_i)_{1 \leq i \leq N}$:

$$\langle f \rangle (t, \boldsymbol{x}) \simeq \frac{1}{N} \sum_{i=1}^{N} f(t, \boldsymbol{x}; \mu_i). \tag{2.11}$$

In the following, and without losing generality, the definition of the ensemble average operator will be not explicitly given. But, we consider a general average operator, which we denote by $\langle \bullet \rangle$ and which satisfies the following properties:

$$\begin{aligned} \langle f + g \rangle &= \langle f \rangle + \langle g \rangle, \\ \langle \theta f \rangle &= \theta \langle f \rangle, \\ \langle \theta \rangle &= \theta, \\ \langle \partial_t f \rangle &= \partial_t \langle f \rangle, \\ \langle \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (f) \rangle &= \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\langle f \rangle), \end{aligned} \tag{2.12}$$

where θ is a scalar coefficient. The first three relations are called the Reynolds rules, the forth is the Leibniz rule, and the fifth is called the Gauss rule.

Other averages can also be used, like the space average or the time average, see Ishii(1975); Drew and Passman(1999) for further details. We underline that these different average operators lead to a similar set of averaged equations, where only the interpretation of the averaged terms can differ.

2.3.2 Governing averaged equations

The local conservative equations (2.2), given for one realization, are valid for a phase k when $\boldsymbol{x} \in \Omega_k$. To write a system of equations valid in the whole domain, we use the characteristic function. By combining both equations (2.2) and (2.5), we can write:

$$\partial_t (\chi_k \rho_k \phi) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\chi_k \rho_k \phi \boldsymbol{u}_k) = \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\chi_k \psi) + \chi_k S + [\rho \phi (\boldsymbol{u} - \mathbf{v}_I) - \psi] \cdot \boldsymbol{\nabla}_{\boldsymbol{x}} (\chi_k). \tag{2.13}$$

Now, by applying the averaging operator defined in the previous section, we obtain the following general form of the averaged equations:

$$\partial_t \langle \chi_k \rho_k \phi \rangle + \nabla_{\boldsymbol{x}} \cdot (\langle \chi_k \rho_k \phi \boldsymbol{u}_k \rangle) = \nabla_{\boldsymbol{x}} \cdot (\langle \chi_k \psi \rangle) + \langle \chi_k S \rangle + \langle [\rho \phi (\boldsymbol{u} - \mathbf{v}_I) - \psi] \cdot \nabla_{\boldsymbol{x}}(\chi_k) \rangle . \quad (2.14)$$

To write the averaged conservative equations in a proper form, we use three types of averaging variables. Let us consider a variable field $f(t, \boldsymbol{x}; \mu)$, where μ again refers to a one given realization. The different averaging operators are defined as follows:

- Phase average:

$$\overline{f}^{\chi_k}(t, \boldsymbol{x}) = \frac{\langle \chi_k f \rangle}{\alpha_k}. \quad (2.15)$$

  where $\alpha_k = \langle \chi_k \rangle$ is the **volume fraction**.

- Favre average:

$$\overline{f}^{\chi_k \rho_k}(t, \boldsymbol{x}) = \frac{\langle \chi_k \rho_k f \rangle}{\alpha_k \overline{\rho_k}^{\chi_k}}. \quad (2.16)$$

- Interfacial average

$$\widetilde{f}^{i}(t, \boldsymbol{x}) = \frac{\langle \boldsymbol{n}_k \cdot \nabla_{\boldsymbol{x}}(\chi_k) f \rangle}{\langle \boldsymbol{n}_k \cdot \nabla_{\boldsymbol{x}}(\chi_k) \rangle}, \quad (2.17)$$

  where $\boldsymbol{n}_k$ is the normal vector at the interface and is oriented towards the other phase $k' \neq k$.

Conservation of mass the averaged mass equation corresponds to $\phi = 1$ in equation (2.14). This equation involves the following averaged quantities: the volume fraction α_k, the averaged mass density $\overline{\rho}_k^{\chi_k}$ and the averaged velocity $\overline{\boldsymbol{u}}_k^{\chi_k \rho_k}$ of the phase k. Then, we can write the averaged mass conservation equation of the phase k as follows:

$$\partial_t \alpha_k \overline{\rho}_k^{\chi_k} + \nabla_{\boldsymbol{x}} \cdot \left(\alpha_k \overline{\rho}_k^{\chi_k} \overline{\boldsymbol{u}}_k^{\chi_k \rho_k} \right) = \Gamma_k, \quad (2.18)$$

where $\Gamma_k = \langle \rho_k [\boldsymbol{u}_k - \mathbf{v}_I] \cdot \nabla_{\boldsymbol{x}}(\chi_k) \rangle$ is the source term due to phase-change, i.e evaporation or condensation. When the phase-change is negelected, this term is equal to zero according to the mass jump condition (2.6). In the general case, we have:

$$\Gamma_l + \Gamma_g = 0. \quad (2.19)$$

This condition ensures the conservation of the total mass of the two phases.

Conservation of momentum the averaged momentum equation corresponds to $\phi = \boldsymbol{u}_k$. The convective part of the equation can be decomposed as follows:

$$\langle \chi_k \rho_k \boldsymbol{u}_k \otimes \boldsymbol{u}_k \rangle = \alpha_k \overline{\rho}_k^{\chi_k} \overline{\boldsymbol{u}}_k^{\chi_k \rho_k} \otimes \overline{\boldsymbol{u}}_k^{\chi_k \rho_k} - \alpha_k \boldsymbol{\tau}_k^t, \quad (2.20)$$

where $\boldsymbol{\tau}_k^t = -\left\langle \chi_k \rho_k (\boldsymbol{u}_k - \overline{\boldsymbol{u}}_k^{\chi_k \rho_k}) \otimes (\boldsymbol{u}_k - \overline{\boldsymbol{u}}_k^{\chi_k \rho_k}) \right\rangle / \alpha_k$ is the Reynolds (fluctuation) stress tensor in the context of two-fluid modeling. It has a similar effect as the Reynolds stress tensor in monophasic Navier-Stokes equations for turbulent flows.

Following the same steps as in Drew and Passman(1999) (chapter 11), the interfacial momentum source term, which corresponds to the last term in the RHS of equation (2.13), can be decomposed as follows:

$$\begin{aligned} \boldsymbol{M}_k &= \langle [\rho_k \boldsymbol{u}_k \otimes (\boldsymbol{u}_k - \mathbf{v}_I) - \boldsymbol{T}] \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \rangle \\ &= \Gamma_k \boldsymbol{u}^m_{i,k} + \widetilde{p_k}^i \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_k) + \boldsymbol{M}^d_k, \end{aligned} \tag{2.21}$$

where $\Gamma_k \boldsymbol{u}^m_{i,k} = \langle \rho_k \boldsymbol{u}_k \otimes (\boldsymbol{u}_k - \mathbf{v}_I) \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \rangle$ is the interfacial momentum source due to phase change, $\widetilde{p_k}^i$ is the interfacial averaged pressure of the phase k and the term $\boldsymbol{M}^d_k$ defines an interfacial force density:

$$\boldsymbol{M}^d_k = \left\langle (p_k - \widetilde{p_k}^i) \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) - \boldsymbol{\tau}_k \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \right\rangle . \tag{2.22}$$

The averaged momentum equation of the phase k can be finally written as follows:

$$\begin{aligned} \partial_t \alpha_k \overline{\rho}^{\chi_k}_k \overline{\boldsymbol{u}}^{\chi_k \rho_k}_k + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\alpha_k \overline{\rho}^{\chi_k}_k \overline{\boldsymbol{u}}^{\chi_k \rho_k}_k \otimes \overline{\boldsymbol{u}}^{\chi_k \rho_k}_k \right) &= -\boldsymbol{\nabla}_{\boldsymbol{x}} \left(\alpha_k \overline{p}^{\chi_k}_k \right) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\alpha_k (\overline{\boldsymbol{\tau}}^{\chi_k}_k + \boldsymbol{\tau}^t_k) \right) \\ &\quad + \Gamma_k \boldsymbol{u}^m_{i,k} + \widetilde{p_k}^i \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_k) + \boldsymbol{M}^d_k . \end{aligned} \tag{2.23}$$

Conservation of energy the averaged energy equation of the phase k, given by $\phi = E_k = e_k + 1/2 \boldsymbol{u}^2_k$, can be derived in a similar way as the previous transport equations. The averaged transport energy equation reads:

$$\begin{aligned} \partial_t (\alpha_k \overline{\rho}^{\chi_k}_k \overline{E}^{\chi_k \rho_k}_k) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\alpha_k \overline{\rho}^{\chi_k}_k \overline{E}^{\chi_k \rho_k}_k \overline{\boldsymbol{u}}^{\chi_k \rho_k}_k \right) &= -\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\alpha_k \left(\overline{\boldsymbol{T}}^{\chi_k}_k \cdot \overline{\boldsymbol{u}}^{\chi_k \rho_k}_k + \overline{\boldsymbol{q}}^{\chi_k}_k + \boldsymbol{q}^t_k \right) \right) \\ &\quad + \Gamma_k E_{k,i} + \Pi_k , \end{aligned} \tag{2.24}$$

where $\boldsymbol{q}^t_k = - \left\langle \chi_k \left(\rho_k (E_k - \overline{E_k}^{\chi_k \rho_k}) \boldsymbol{I} + \boldsymbol{T} \right) \cdot (\boldsymbol{u}_k - \overline{\boldsymbol{u}_k}^{\chi_k \rho_k}) \right\rangle / \alpha_k$ is the total fluctuation energy flux. This term can be decomposed into different fluctuation terms (internal energy, kinetic energy and shear fluctuations), as was suggested in Drew(1983). The term $\Gamma_k E^m_{k,i} = \langle \rho_k E_k (\boldsymbol{u}_k - \mathbf{v}_I) \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \rangle$ is the interfacial energy source term due to phase change. And finally, $\Pi_k = \langle (\boldsymbol{T} \cdot \boldsymbol{u}_k + \boldsymbol{q}) \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \rangle$ is the sum of interfacial work and heat source terms.

In the following, we omit the bar notation used for the average quantities. We also suppose one interface velocity $\mathbf{v}_I = \widetilde{\boldsymbol{u}_k}^i$ and one interface pressure $p_I = \widetilde{p_k}^i$ for both phases. In the first equality, the effect of the phase change on the interface velocity is neglected. While for the second one, we neglect the surface tension effect on the interface pressure. We underline that these assumptions do not require any equilibrium assumption between the velocity or the pressure of the two phases.

Finally, we obtain three equations given in (2.18), (2.23) and (2.24) for each phase. These equations are written here in a general form. Indeed, no assumptions on the interface topology and the equilibrium between the phases have been yet used. The obtained system of equations is a non-conservative system because of the product terms of the interface variables (velocity and pressure) and the gradient of the volume fraction. These terms condition the mathematical structure of the equations and they require specific treatments for their numerical resolution Saurel and Abgrall(1999).

2.3.3 Volume fraction transport equation

In this section, we consider compressible flows. In the case where the flow is incompressible, the derivation of the volume fraction equation is straightforward from the mass conservation equation (2.18). But, the compressible case is more complicated. Indeed, when we apply the average operator on the kinematic equation of the presence function (2.5), we obtain:

$$\partial_t \langle \chi_k \rangle + \langle \boldsymbol{u}_k \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \rangle = 0. \tag{2.25}$$

So far, we do not know how to express properly the term $\langle \boldsymbol{u}_k \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k) \rangle$ as a differential term of the averaged variables of the problem. Consequently, some contributions propose to use a postulated equation for the volume fraction Bergles et al.(1981) or by deriving the transport equation from some thermodynamic considerations. The second approach is more consistent with the second principle of thermodynamics. Baer and Nunziato Baer and Nunziato(1986) were the first to derive the equation on α based on this approach. The resulting equation reads:

$$\partial_t \alpha_k + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_k) = S_\alpha, \tag{2.26}$$

where $\mathbf{v}_I$ is a mean interface velocity, which is often modeled using the two phases velocities and S_α is a source term which is often written as a pressure relaxation term $S_\alpha = \mu(p_k - p_{k'})$ and μ is a parameter of the model. This relaxation represents the rate of expansion of the volume fraction to tend the pressures towards equilibrium as pointed by Saurel and Metayer(2000). Even if equation (2.26) is widely used in many two-phase flows modeling, we underline that its interpretation is still misunderstood in the case of a dilute disperse phase Lhuillier et al.(2013).

2.4 Closures and Classifications

The averaged conservative equations, presented in the previous section, are given in a general unclosed form. At this level, we have six transport equations (three for the liquid and three for the gas) or seven equations when adding a transport equation on the volume fraction. The unclosed terms need be modeled and expressed as a function of the variables of the system $(\alpha_l, \rho_k, \boldsymbol{u}_k, E_k)$. These terms can be classified as follows:

A. unclosed thermodynamic terms can be closed through an appropriate Equation of State (EoS). For example, a pressure law $p_k(\rho_k, e_k)$ needs to be specified for each material k.

B. the volume fraction α_k is used in this chapter as a variable of the problem. In other class of models, this term is not transported and it can be obtained by using a mechanical equilibrium assumption between the gas and the liquid pressures, see Drui et al.(2016b) and related references:

$$p_l(m_l/\alpha_l, e_l) = p_g(m_g/(1-\alpha_l), e_g), \tag{2.27}$$

where $m_k = \alpha_k \rho_k$.

C. fluctuation terms $\boldsymbol{\tau}_k^t$and $\boldsymbol{q}_k^t$ are related to the fluctuation part of the transport variables. The modeling of these terms is often inspired from turbulent modeling works, where the fluctuation tensors are expressed as additional diffusion terms in the system of equations Devassy et al.(2015).

D. interfacial quantities p_I and $\mathbf{v}_I$, are involved in the conservative equations as multipliers of gradient terms. Consequently their expression conditions the mathematical structure of the system and the wave dynamics in the flow. One should be careful in modeling these terms in a way that ensures good mathematical properties. Indeed, it is important to express the interfacial variable as a function of the problem variables in a way that the convective part of the system of equations is hyperbolic and it satisfies an additional entropy equation in the case of regular solutions Seguin(2002). This will be discussed further in the following section.

E. exchange source terms between the two phases Γ_k, $\Gamma_k \boldsymbol{u}^m_{k,i}$, $\Gamma_k E^m_{k,i}$, $\boldsymbol{M}^d_k$ and Π_k need to be modeled according to the physical process and the studied phenomenon. Nevertheless, these source terms should be dissipative, in order to satisfy the second principle of thermodynamics Baer and Nunziato(1986); Guillemaud(2007b); Guillemaud(2007a). The fulfillment of this condition provides admissible closure for the exchange terms as stated by Baer and Nunziato(1986).

So far, we have seven equations (three for each phase) and a transport equation on the volume fraction. In the literature, other two-fluid/mixture[2] models involves different number of averaged equations. In general, this number goes from seven equations down to three, depending on the nature of the equilibrium expected between the two phases: the more equilibrium the smallest number of equations. The equilibrium assumption can be classified as follows:

- kinematic equilibrium: $\boldsymbol{u}_g = \boldsymbol{u}_l$,
- mechanical equilibrium: $p_g = p_l$,
- thermal equilibrium: $T_g = T_l$,
- the thermodynamic equilibrium corresponds to both a mechanical and a thermal equilibrium.

The seven-equation model supposes a full disequilibrium between the phases. A first model of this type was proposed by Baer and Nunziato Baer and Nunziato(1986). The mathematical properties of the model was then studied in different works Saurel and Abgrall(1999); Gavrilyuk and Saurel(2002); Gallouet et al.(2004); Saurel et al.(2009); Ambroso et al.(2009). In the case where the diffusion terms, the external source terms (body force, radiation) and the phase change are neglected, this model reads:

$$\begin{array}{rcl}
\partial_t \alpha_l + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_l) & = & \mu(p_l - p_g), \\
\partial_t \alpha_k \rho_k + \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_k \rho_k \boldsymbol{u}_k) & = & 0, \\
\partial_t \alpha_k \rho_k \boldsymbol{u}_k + \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_k(\rho_k \boldsymbol{u}_k \otimes \boldsymbol{u}_k + p_k \boldsymbol{I})) & = & p_I \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_l) + \lambda(\boldsymbol{u}_k - \boldsymbol{u}_{k'}), \\
\partial_t \alpha_k \rho_k E_k + \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_k(\rho_k E_k + p_k)\boldsymbol{u}_k) & = & p_I \mathbf{v}_I \boldsymbol{\nabla}_{\boldsymbol{x}}(\alpha_l) + \lambda \mathbf{v}_I \cdot (\boldsymbol{u}_k - \boldsymbol{u}_{k'}) - p_I \mu(p_k - p_{k'}).
\end{array} \tag{2.28}$$

The coefficients λ and μ are relaxation coefficients, which derive the system to equilibrium for infinite values. When one or two of these variables tend to infinity, we can reduce the number of equations and it is possible to derive some other models by this way. For example by letting λ and μ go to infinity and supposing a thermal equilibrium between the two phases, one can show that the seven-equation model can degenerate to the Homogeneous Equilibrium Model (HEM) Barberon and Helluy(2005); Faccanoni et al.(2012); Drui et al.(2016b); Hurisse(2017).

---

[2] An averaged two-phase flow model is considered as mixture model when we consider one velocity for the two phases.

2.5 Mathematical structure of the equations

In the following, we discuss some mathematical properties of the convective part of the averaged two-phase flow models. A wide range of these models can be found in the literature. They can differ in numerous ways: the number of equations, the equilibrium assumptions between the phases, the closure relations of the interfacial variables and the considered EoS. The objective is not to give an exhaustive analysis of all two-fluid models. However, we present some conditions, especially on the interfacial variables, which ensure good mathematical properties of the model. We restrict our study to the convective part of the equations. For the sake of simplicity, we consider a 1D domain. In general, the convective part of two-fluid models can be written in 1D in the following form:

$$\partial_t \boldsymbol{U} + \partial_x \left(\boldsymbol{\mathcal{F}}\left(\boldsymbol{U}\right)\right) + B(\boldsymbol{U})\partial_x \boldsymbol{U} = 0, \tag{2.29}$$

where $\boldsymbol{U} \in \mathbb{R}^m$ is the vector of transported variables, the map $\boldsymbol{\mathcal{F}} : \mathbb{R}^m \rightarrow: \mathbb{R}^m$ is the flux function, $B(\boldsymbol{U})$ is $m \times m$ matrix, which mainly depends on the intefacial variables and $3 \leq m \leq 7$ is the number of transported equations that depends on the equilibrium assumptions between the two phases. In the case of the model given in (2.28) ($m = 7$), these variables can be expressed as follows:

$$\boldsymbol{U} = \begin{pmatrix} \alpha_l \\ \alpha_l \rho_l \\ (1-\alpha_l)\rho_g \\ \alpha_l \rho_l u_l \\ (1-\alpha_l)\rho_g u_g \\ \alpha_l \rho_l E_l \\ (1-\alpha_l)\rho_g E_g \end{pmatrix}, \quad \boldsymbol{\mathcal{F}}\left(\boldsymbol{U}\right) = \begin{pmatrix} 0 \\ \alpha_l \rho_l u_l \\ (1-\alpha_l)\rho_g u_g \\ \alpha_l \rho_l u_l^2 + \alpha_l p_l \\ (1-\alpha_l)\rho_g u_g^2 + (1-\alpha_l) p_g \\ \alpha_l(\rho_l E_l + p_l)u_l \\ (1-\alpha_l)(\rho_g E_g + p_g)u_g \end{pmatrix} \tag{2.30}$$

and

$$B\left(\boldsymbol{U}\right) = \begin{pmatrix} v_I, & 0, & \dots, & 0 \\ 0, & 0, & \dots, & 0 \\ 0, & 0, & \dots, & 0 \\ -p_i, & 0, & \dots, & 0 \\ p_i, & 0, & \dots, & 0 \\ -p_i v_i, & 0, & \dots, & 0 \\ p_i v_i, & 0, & \dots, & 0 \end{pmatrix} \tag{2.31}$$

Considering regular solutions of the system (2.29), we can write the convective part in a quasi-linear form:

$$\partial_t \boldsymbol{U} + A(\boldsymbol{U})\partial_x \boldsymbol{U} = 0, \tag{2.32}$$

where $A(\boldsymbol{U}) = \boldsymbol{\nabla}_{\boldsymbol{U}}(\boldsymbol{\mathcal{F}}\left(\boldsymbol{U}\right)) + B\left(\boldsymbol{U}\right)$ is a $m \times m$ matrix. The system is hyperbolic if the matrix $A(\boldsymbol{U})$ is diagonalizable and all their eigenvalues are reals. Physically, this means that the waves of the system travel at defined real velocities. In the case where one of the eigenvalues has an imaginary part, the system leads to some singularities Ndjinga(2007). Consequently, the model is mathematically an ill-posed model. Another important property of the system of equations is the existence of an entropy equation. In general, an entropy is a convex function $\eta(\boldsymbol{U})$ of the variables of the system. In the case of the existence of regular solution of the original system (2.29), this quantity satisfies a transport equation in the following form:

$$\partial_t \eta + \partial_x f(\eta) = 0, \tag{2.33}$$

where $f(\eta) \in \mathbb{R}$ is the entropy flux.

In the case of the seven-equation model of Baer & Nunziato, the system of equations has an entropy equation given by the entropy $\eta = \sum_k \alpha_k \rho_k s_k$ and its flux $f(\eta) = \sum_k \alpha_k \rho_k s_k u_k$, where s_k is the physical entropy of the material k that satisfies the Gibbs relation:

$$ds_k = \frac{de_k}{T_k} + \frac{p_k}{T_k} d\left(\frac{1}{\rho_k}\right). \tag{2.34}$$

In the case of weak solutions[3] of the original system of equations (2.29), the entropy equation becomes an inequality $\partial_t \eta + \partial_x f(\eta) \geq 0$ Godlewski and Raviart(1996). For a given initial and boundary conditions for which a weak solution exists, the entropic inequality allows to define a unique weak solution. The hyperbolic and the existence of an entropy equation are two important mathematical properties that can help in the well-posedness of the model. Consequently, robust numerical schemes are more likely to provide stable simulations.

The different assumptions that one can consider such as compressibility effects, equilibrium assumptions between the two phases and closure relations of the interface pressure and velocity have a direct impact on the hyperbolicity of the obtained equations and the existence of an entropy equation. It is important to preserve these properties, when we model the two phases. In fact, a good physical model is most likely to have good mathematical properties as it was pointed by Drew and Passmann Drew and Passman(1999). However, some common assumptions used in two-fluid modeling can lead to a non-hyperbolic system. Let us give two examples. First, it is very common to suppose that the liquid phase is incompressible and that the interface pressure is imposed by the gas pressure $p_I = p_g = p$. This modeling assumption seems to be physically reasonable and justified by the fact that the liquid has a very low compressibility and a very short time-scale of the pressure relaxation toward an equilibrium pressure, compared with other time scales. Nonetheless, it yields a non-hyperbolic system of equations and thus an ill-posed mathematical model Saurel and Abgrall(1999). Another ill-posed model occurs when a mechanical equilibrium between the two thermodynamic pressures $p_l(\rho_l, e_l) = p_g(\rho_g, e_g)$ is imposed. This type of model is known as the six-equation model (mass, momentum and energy), while the volume fraction is obtained from the pressure equilibrium assumption. The six-equation model has been shown to be non-hyperbolic in Boure(1997). Yet, this model was widely used in nuclear engineering. We refer here to the work of Ishii & Hibiki Ishii and Hibiki(2011) and Morel Morel(1997). Their choice was justified by the fact that the propagation of sound waves is not paramount in their studies. Plus, they use an additional diffusion terms, which stabilize the numerical resolution. In the opposite, Saurel and Abgrall(1999) showed that, when both phases are considered as compressible with two different pressures, the system of equations has more degrees of freedom and it is possible to obtain hyperbolic model provided that we use consistent closures of the interface variables. Another point that affects the hyperbolic property and the nature of the system waves is the interface variables. Saurel and Metayer(2000) and Delhaye and Boure(1982) suggest the following expressions:

$$\begin{aligned} p_I &= \sum_{k=l,g} \alpha_k (p_k + \rho_k (v_I - u_k)^2), \\ v_I &= \sum_{k=l,g} \alpha_k \rho_k u_k / \sum_{k=l,g} \alpha_k \rho_k, \end{aligned} \tag{2.35}$$

These relations lead to a hyperbolic system. Indeed, by using these relations in the system (2.28), we can show in 1D that the seven-equation model has seven eigenvalues $(u_k, v_I, u_k \pm c_k)$, where c_k is the speed of sound in phase k. The eigenvalues are distinct, thus we have a strictly hyperbolic

---

[3] weak solution is defined as the solution of the differential equations in the sense of distributions.

system, except when locally some eigenvalues are degenerate. Furthermore, the second equation of (2.35) ensures that the characteristic field associated with v_I is linearly degenerates, which allows to simplify the derivation of an approximate solution of Riemann problem Saurel and Abgrall(1999).

The fundamental conclusion of this section is that additional relations can simplify the modeling but it can also lead to some mathematical difficulties. For this reason, one needs to be aware of these issues in order to be able to propose a physically consistent model but also a mathematically well-posed system of equations.

2.6 Interface sharpening methods

So far, we have discussed the physical consistency and the mathematical properties of two-fluid models. However, their accuracy has not yet been considered. As explained before, two-fluid models can spread the interface by including, in the source terms of the model, some fluctuation effects on the flow. The diffusion process of the two-fluid model interface is interpreted as the averaged evolution across multiple realizations of the real interface. It is important that the source terms can reproduce correctly this process, when the fluctuations and the interface instabilities are very high. Otherwise, we need to preserve a sharp interface. However, classical numerical methods produce artificial diffusion of contact discontinuities, resulting in an artificial fluid mixing at the interface. This numerical diffusion is not a physical process. As consequence, it will destroy the interface, in a region where the real interface is not necessary deformed. Furthermore, in the region where the interface is very deformed, the numerical diffusion will lead to overestimate these deformations. For this reason, it is important to counter the numerical diffusion of the interface and to preserve the large interface structures, especially near the nozzle outlet. In the literature, different contributions have addressed this problem in different manners. First, high order numerical schemes generally decrease the numerical diffusion of the solution. This works well for regular solutions. In the presence of the interface, it is not obvious to preserve a sharp interface, while ensuring a non oscillating solution. Another possibility consists in using a locally high grid resolution. This can be achieved using Adaptive Mesh Refinement (AMR) codes, see Drui et al.(2016a); Drui(2017) and references therein. A third option, to cope with this problem, is to use anti-diffusive numerical schemes. We refer here to the works So et al.(2012) and Kokh and Lagoutière(2010), where the method has been used for a sharp two-phase flow model[4]. However, these methods rely on the underlying numerical scheme as it was pointed by Tiwari et al.(2013). Furthermore, some anti-diffusion methods So et al.(2012) can over-sharpen the interface and lead to some numerical instabilities. Finally, additional source terms can be envisioned to regularize the interface and keep an approximately constant-thickness interface Tiwari et al.(2013); Shukla et al.(2010). But, these source terms can be incompatibles with the thermodynamics of the two-fluid model. Tiwari et al.(2013) have developed a consistent interface regularization source term. In their work, they show some promising results using a compressible bubble collapse test case. They compared their results with other methods that regularize the interface So et al.(2012); Shukla et al.(2010) and with a semi-analytic solution. The result obtained by their method shows a good evaluation of the bubble compression and a good preservation of the interface-thickness. Although, it is important to point out that many of these methods depend on some parameters which need to be initially determined by the user. It will be worthwhile, in future work, to relate these parameters to some relevant variables of

---

[4] the five-equation model used in Kokh and Lagoutière(2010) diffuses the interface only numerically, for this reason the model is consider as a sharp interface model. This model does not belong to the class of models that we present in the present chapter.

the model.

2.7 Sub-scale interface modeling

Two-fluid models are often restricted to describe the interface by one interface function, which is here given by the volume fraction variable. This variable varies smoothly in the interface zone because of the artificial diffusion. The regularization interface methods, discussed in the previous section, can success in limiting the diffusion in an approximately constant-thin zone. The thickness of this zone is often given by few computational cells. Therefore, even in optimistic cases, we are not able to capture the interface at scales smaller than the mesh size. This problem is more relevant in the intermediate zone where the atomization occurs and in the disperse phase zone. These zones present very small interface scales compared to coarse meshes. In order to recover some interface topology information, some additional topological variables can be transported. In the modeling of the primary atomization, many contributions Devassy et al.(2015); Lebas et al.(2009); Morel(1997) have used an additional transport equation for a surface area density variable. Drew(1990) proposed to go further in describing the interface by using two interface curvatures in addition to the volume fraction and the surface area density. The derivation of transport equations of these geometrical interface variables is not straightforward. To illustrate this difficulty, the transport equation of a surface area density will be presented in the following. The curvature variables are the subject of the second part of this manuscript and will not be presented in this chapter.

Area density transport equation The surface area density is defined as the mean interfacial area per unit of volume. This is an important quantity to better account for the interface topology and for an accurate modeling of the exchange transfer terms at the interface. Furthermore, this quantity and the volume fraction provide an estimation of a mean droplet size in a disperse phase flow $d = \Sigma_I/(6\alpha)$, where Σ_I is the surface area density. We can express this quantity, in the context of ensemble average, as follows:

$$\Sigma_I = \langle \delta_I(t,\boldsymbol{x}) \rangle \,, \tag{2.36}$$

where

$$\delta_I(t,\boldsymbol{x}) = ||\boldsymbol{\nabla}_{\boldsymbol{x}}(\chi_k(t,\boldsymbol{x}))||, \tag{2.37}$$

is the instantaneous area concentration. We underline that the derivation in (2.37) is defined in the sense of distributions. The instantaneous area concentration (IAC) was introduced in the first time in Kataoka(1986), where the surface of the gas-liquid interface is defined as zeros of a function $g(\boldsymbol{x},t) = 0$. In this context, IAC was defined as follows:

$$\delta_I(t,\boldsymbol{x}) = ||\boldsymbol{\nabla}_{\boldsymbol{x}}(g)||\delta(g(t,\boldsymbol{x})). \tag{2.38}$$

Both definitions (2.37) and (2.38) are in fact equivalents. We can show this result by using the fact that $\chi_k = \mathcal{H}(g(t,\boldsymbol{x}))$ or $\chi_k = \mathcal{H}(-g(t,\boldsymbol{x}))$, where $\mathcal{H}$ is the Heaviside function. In Morel(2015), a transport equation for the IAC is derived:

$$\partial_t \delta_I + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\delta_I \mathbf{v}_I) = \delta_I \boldsymbol{\nabla}_s \cdot (\mathbf{v}_I), \tag{2.39}$$

where $\boldsymbol{\nabla}_s \cdot (\bullet) = (\boldsymbol{I} - \boldsymbol{n} \otimes \boldsymbol{n}) : \boldsymbol{\nabla}_{\boldsymbol{x}}(\bullet)$ is the surface divergence operator. The averaged transport equation of the surface area density can be obtained by applying the averaging operator:

$$\partial_t \Sigma_I + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\Sigma_I \mathbf{v}_I) = \langle \delta_I \boldsymbol{\nabla}_s \cdot (\mathbf{v}_I) \rangle \,. \tag{2.40}$$

This equation does not take into account discontinuous phenomena, like the breakage or the coalescence. To take these phenomena into consideration, we need to add some source terms in this equation Morel(2015). The term in the RHS is the surface stretching source term. This term needs also to be closed and related to the transported variables. Other transport equations for the surface area density have been proposed in the literature see Vallet et al.(2001); Devassy et al.(2015); Lebas et al.(2009) and references therein. The derivation of transport equation is based on phenomenological modeling of the interface stretching and diffusion due to interface fluctuation:

$$\partial_t \Sigma_I + \nabla_{\boldsymbol{x}} \cdot (\Sigma_I \boldsymbol{u}) = \nabla_{\boldsymbol{x}} \cdot (D_s \nabla_{\boldsymbol{x}} (\Sigma_I)) + (A + a)\Sigma_I - V_s \Sigma_I, \tag{2.41}$$

where D_s is a diffusion coefficient, $1/A$ and $1/a$ are two different production time scales and V_s is a destruction coefficient homogeneous to a velocity.

We also underline that the concept of the surface area density was used in combustion field Candel and Poinsot(1990).

2.8 Conclusion

Two-fluid models can be derived from a statistical average approach. This class of models is suitable to simulate the separated two-phase flows, where the details on the gas-liquid interface are not completely resolved. In the literature a wide range of two-fluid models are considered. Each model depends on some equilibrium assumptions and closure relations. This chapter presents paramount elements that can help to choose or to derive a mathematically and physically consistent model. First, it is important to satisfy some mathematical properties, such as the hyperbolicity of the convective part of the system of equations and the existence of an entropy equation. Furthermore, the modeling of the source terms should be globally dissipative to satisfy the second law of thermodynamics. Finally, the source terms should include the fluctuations and the sub-scale effects on the mean flow. While, the numerical diffusion of the interface should be countered and limited using consistent methods.

This brief review on two-fluid models highlights a crucial difficulty in modeling the interface. Indeed, the statistical description of the interface through the volume fraction is still very poor to take into account complex interface topology. An accurate description of the interface subscale is important to model the primary atomization. Consequently, two-fluid models are not yet mature to describe correctly the whole process of fuel injection. The main topic of this PhD consists in enriching the interface by introducing new interface variables. But before we present our approach to this issue, in the following chapter, we first present an overview of the existing Eulerian moment models to describe the disperse phase.

Chapter 3

Eulerian modeling of spray

3.1 Introduction

In the previous chapter, we showed that two-fluid models can be used to model separated two-phase flows. This type of flow is present close to the outlet nozzle of fuel injectors. Additional transport equations like a surface area-density transport equation can be used to enhance the gas-liquid interface description and to capture the primary atomization in the intermediate injection zone. However, even by using this variable, the current two-fluid models fail to accurately simulate the flow in the downstream region. Indeed, the atomization of the liquid fuel yields to a polydisperse evaporating spray. This spray consists of a cloud of small droplets of different sizes, carried by the continuous gaseous phase. The volume fraction and the surface area density variables can only estimate a mean diameter of the droplets, while the size distribution remains unknown. For combustion applications, it is important to capture the evaporation and the mixing of the fuel with the oxidant (air). These mechanisms are mainly controlled by the polydispersion character of the spray (see Figure 3.1) and also by the transport of the gas species. Actual two-fluid models are not yet able to describe accurately this polydispersion. Furthermore, there is no model that can describe and simulate accurately the two zones of the flow: disperse and separated phases. Therefore, in spray-combustion simulations, we often consider that the boundary conditions are moved downstream into the disperse region, and the spray is injected with an initial distribution (without taking into account the primary atomization process). This distribution is mainly obtained from experimental data. In this context, the gaseous phase is simulated on an Eulerian grid, while droplets can be simulated using Lagrangian Apte

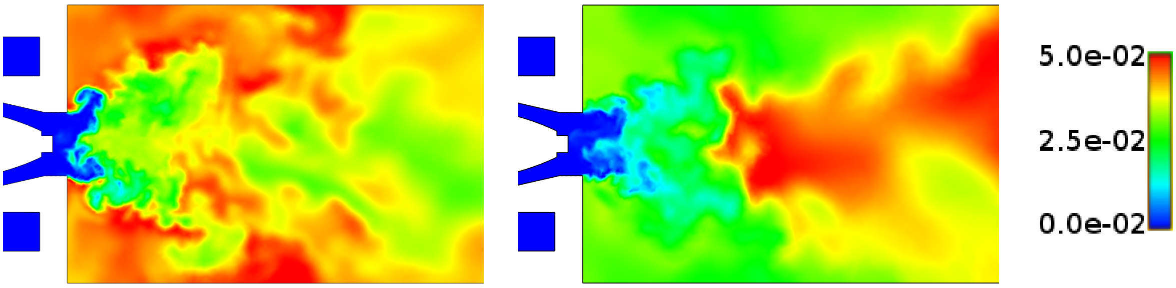


Figure 3.1: *Simulation of the MERCATO test rig of ONERA Vié et al.(2013a). Vapor fuel mass fraction obtained for monodisperse spray (left) and for a polydisperse spray using multi-fluid model (right) for non-reactive conditions.*

et al.(2003); Sanjosé et al.(2011); Cunha(2018) or Eulerian models Martinez et al.(2010); Emre et al.(2015); Kah et al.(2015); Vié et al.(2015b). In this chapter, we present a general overview of the Eulerian spray models based on the moments method. For these models, we consider the gas mixture as the carrier and continuous phase, described as a monophasic flow. The spray consists of disperse droplets of presumed interface shape (spherical droplet), which can be described accurately with a kinetic approach Williams(1958). Starting from this kinetic equation, reduced models can be derived using the moments method. Eventually, the interaction between the two phases is taken into account through additional source terms.

3.2 Gaseous phase description

3.2.1 Governing equations

As mentioned in the introduction, accurate description of the gaseous phase mixture is an important element for an accurate prediction of the combustion. Indeed, the reaction rate depends on the local mass fraction of each species involved in the different intermediary chemical mechanisms. Such relation can be derived from kinetic chemical considerations Poinsot and Veynante(2005). Navier-Stokes equations for a multi-species reactive flow can be used to simulate the continuous gaseous phase. In the following, we suppose that the disperse phase occupies locally a small volume fraction, such that its interaction with the gaseous phase can be modeled simply through source terms in Navier-Stokes equations. The Eulerian equations for the gas phase can be obtained by expressing mass, momentum and energy balances on a control volume Candel(1990):

$$
\begin{aligned}
\partial_t \rho_g + \nabla_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g\right) &= S^m, \\
\partial_t \left(\rho_g Y_k\right) + \nabla_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g Y_k\right) &= -\nabla_{\boldsymbol{x}} \cdot \left(\rho_g Y_k \boldsymbol{u}_k^{diff}\right) + \dot{\omega}_k + S_k^{species}, \\
\partial_t (\rho_g \boldsymbol{u}_g) + \nabla_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g \otimes \boldsymbol{u}_g\right) &= -\nabla_{\boldsymbol{x}} \left(P_g\right) + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}) + S^{mom}, \\
\partial_t (\rho_g h_g^s) + \nabla_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g \, h_g^s\right) &= -\nabla_{\boldsymbol{x}} \cdot (\boldsymbol{q}) + \partial_t (P_g) + \boldsymbol{\tau} : \nabla_{\boldsymbol{x}} \left(\boldsymbol{u}_g\right) + \dot{\omega}_T + S^{enth}.
\end{aligned}
\tag{3.1}
$$

where ρ_g is the mass density of the gas mixture, $\boldsymbol{u}_g$ is the gas velocity, h_g^s is the sensible enthalpy of the mixture and P_g is the mixture gas pressure. These variables characterize the global mixture of the gas and depend on its composition, which is described by the mass fraction $Y_k = \rho_k / \rho_g$ of each species k, and where ρ_k is the mass density of the species k. The global mixture quantities satisfy:

$$
\sum_{k=1}^{n_{sp}} \rho_k = \rho_g, \qquad \sum_{k=1}^{n_{sp}} Y_k \boldsymbol{u}_k = \boldsymbol{u}_g \quad \text{and} \quad \sum_{k=1}^{n_{sp}} Y_k h_k^s = h_g^s. \tag{3.2}
$$

where $\boldsymbol{u}_k$ is the velocity of species k. Since the species velocity is not necessarily equal to the mixture gas velocity, the relative velocity $\boldsymbol{u}_k^{diff} = \boldsymbol{u}_k - \boldsymbol{u}_g$ is the diffusion velocity of the species k within the gas mixture. The sensible enthalpy $h_k^s = h_k - h_k^0$ of the species k is defined as the difference of the total enthalpy h_k of the species k and the enthalpy of formation h_k^0 defined at a standard pressure and temperature. The sensible enthalpy of the gas mixture can be expressed using the heat capacity of the gas mixture $C_{p,g}(T) = \sum_{k=1}^{n_{sp}} Y_k C_{p,k}(T)$ as follows:

$$
h_g^s = \int_{T_0}^{T_g} C_{p,g}(T) \, dT. \tag{3.3}
$$

The right hand side of the system (3.1) corresponds to the evaporation and drag source terms of the disperse phase, the chemical reactions of the species and some diffusion terms due to the species diffusion velocity, dynamic viscosity and thermal fluxes. The source terms related to the disperse phase are: S^m the mass source term due to the spray evaporation, S^{mom} the momentum source due to the evaporation and drag force of the disperse phase and S^{enth} energy source term due to the disperse phase. Then, the other source terms are: $S_k^{species}$ the evaporation source term of the species k, $\dot{\omega}_k$ the reactive rate due to chemical reaction of the species k, $\boldsymbol{\tau}$ the viscous stress tensor, $\dot{\omega}_T$ the heat release source term due to chemical reaction and $\boldsymbol{q}$ the thermal flux.

3.2.2 Closure models

The system of equations (3.1) consists of $2 + n_{sp} + d$ equations, where d is the space dimension. The unknown variables to be solved are the mass density ρ_g, the velocity $\boldsymbol{u}_g$, the sensible enthalpy of the gas mixture h_g^s and the mass fraction Y_k of each species. The system of equations needs some closure assumptions and additional relations to relate the different terms to the transported variables of the problem. The unclosed source terms due to the disperse phase will be discussed later in this chapter. The source terms due to the chemical reactions depend on the reactive rate of these reactions. For this reason, one needs to conduct chemical kinetic studies to determine the different intermediate reaction steps and identify the preponderant reactions. Arrhenius law allows to estimate the reaction rate after identifying the important intermediate reaction steps. If we neglect the diffusion of the mass due to temperature gradients, known also by the Soret effect, we can then use the Fick's law:

$$Y_k \boldsymbol{u}_k^{diff} = -D_k \, \partial_{\boldsymbol{x}} \cdot (Y_k) \tag{3.4}$$

where D_k is the diffusion coefficient of species k into the mixture. In the momentum equation, we consider a Newtonian fluid to express the viscous stress tensor:

$$\boldsymbol{\tau} = \mu_g \left(\boldsymbol{\nabla}_{\boldsymbol{x}} \left(\boldsymbol{u}_g \right) + \boldsymbol{\nabla}_{\boldsymbol{x}} \left(\boldsymbol{u}_g \right)^t \right) - \frac{2}{3} \mu_g \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\boldsymbol{u}_g \right) \boldsymbol{I}, \tag{3.5}$$

where μ_g is the dynamic viscosity of the gas.

The heat release due to the combustion chemical reactions can be computed after determining the chemical reaction rate of each species as follows:

$$\dot{\omega}_T = \sum_{k=1}^{n_{sp}} \dot{\omega}_k h_k^0. \tag{3.6}$$

The energy flux can be modeled by a Fourier's law and a term associated with different enthalpy species diffusion:

$$\boldsymbol{q} = -\lambda_g \partial_{\boldsymbol{x}} \, T_g + \rho_g \sum_{k=1}^{n_{sp}} h_k \, \boldsymbol{u}_k^{diff} Y_k. \tag{3.7}$$

Equations (3.4)-(3.7) close the source terms of system (3.1) excepted the source terms related to the spray. In the case of a one way-coupling, these terms are neglected, and thus the system is fully closed. But in such case, we cannot simulate gaseous fuel production through evaporation. Accurate modeling of source terms due to the disperse phase will be presented later in this chapter, where a two-way coupling model framework is considered. This second approach is more accurate to describe the gas-spray interactions.

3.2.3 Dimensionless variables

In this section, we present the dimensionless equations of the gaseous phase and we introduce in this context the different dimensionless variables that control the flow regime. First, let us introduce the characteristic variables: U_0, the reference macroscopic velocity and x_0, the reference length of the domain, which allow to define a reference time scale for the gas: $\tau_g = x_0/U_0$. These quantities, along with the physical constants for a reference physical mixture, ρ_∞, μ_∞, $C_{p,\infty}$, T_∞, W_∞ are taken to define the dimensionless system. The reference pressure is defined by $P_\infty = \rho_\infty R T_\infty / W_\infty$. For any variables a, its dimensionless value $\overline{a}$ is given by $\overline{a} = a/a_0$ or $\overline{a} = a/a_\infty$. To derive the dimensionless equations, we define a normalization Reynolds number based on the reference quantities:

$$\mathrm{Re}_0 = \frac{\rho_\infty \; x_0 \; U_0}{\mu_\infty}. \tag{3.8}$$

The molecular transports of species and heat are characterized by the following dimensionless variables Poinsot and Veynante(2005):

$$\mathrm{Sc}_k = \frac{\mu_g}{\rho_g D_k}, \quad \mathrm{Pr} = \frac{C_{p,g}\mu_g}{\lambda_g}, \quad \mathrm{Le}_k = \frac{\lambda_g}{\rho_g \; C_{p,g} \; D_k}. \tag{3.9}$$

The Schmidt number Sc_k for the species k compares momentum dissipation and species k molecular diffusion, whereas the Prandtl number Pr compares the thermal diffusivity $\lambda_g/\rho_g C_{p,g}$ to the mechanical diffusivity μ_g/ρ_g. The mass and thermal diffusivity are compared via the Lewis number $\mathrm{Le}_k = \mathrm{Sc}_k/\mathrm{Pr}$. In order to obtain a simple form of the dimensionless system of equations, similar to the original one, we introduce the quantities:

$$\mu_g^\star = \frac{\overline{\mu_g}}{\mathrm{Re}_0}, \quad D_k^\star = \frac{\overline{\mu_g}}{\overline{\rho_g} \; \mathrm{Re}_0 \; \mathrm{Sc}_k}, \quad \lambda_g^\star = \frac{\overline{C_{p,g}} \; \overline{\mu_g}}{\mathrm{Re}_0 \; \mathrm{Pr}}. \tag{3.10}$$

With these definitions, the Schmidt, Prandtl and Lewis numbers are given by:

$$\mathrm{Sc}_k = \frac{\mu_g^\star}{\overline{\rho_g} \; D_k^\star}, \quad \mathrm{Pr} = \frac{\overline{C_{p,g}} \; \mu_g^\star}{\lambda_g^\star}, \quad \mathrm{Le}_k = \frac{\lambda_g^\star}{\overline{\rho_g} \; \overline{C_{p,g}} \; D_k^\star}. \tag{3.11}$$

In this framework, we can write the gaseous system in non dimensional form, obtained from equation (3.1). To lighten the notations, we use for dimensionless variables, the same variables as for the original ones:

$$\begin{aligned}
\partial_t \, \rho_g + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g\right) &= S^m, \\
\partial_t \left(\rho_g Y_k\right) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g Y_k\right) &= \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\rho_g D_k^\star \, \boldsymbol{\nabla}_{\boldsymbol{x}}(Y_k)\right) + \dot{\omega}_k + S_k^{species}, \\
\partial_t(\rho_g \boldsymbol{u}_g) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g \otimes \boldsymbol{u}_g\right) &= -\frac{1}{\gamma_\infty M^2} \boldsymbol{\nabla}_{\boldsymbol{x}}\left(P_g\right) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}) + S^{mom}, \\
\partial_t(\rho_g h_s) + \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\rho_g \boldsymbol{u}_g \; h_s\right) &= -\boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\boldsymbol{q}^\star) + \frac{\gamma_\infty - 1}{\gamma_\infty} \, \partial_t(P_g) \\
&\quad + M^2(\gamma_\infty - 1) \; \boldsymbol{\tau} : \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\boldsymbol{u}_g\right) + \dot{\omega}_T + S^{enth},
\end{aligned} \tag{3.12}$$

with

$$\boldsymbol{q}^\star = -\lambda_g^\star \, \partial_{\boldsymbol{x}} \, T_g + \rho_g \sum_{k=1}^{n_{sp}} h_k \; D_k^\star \; \boldsymbol{\nabla}_{\boldsymbol{x}}(Y_k). \tag{3.13}$$

We define the reference Mach number M, and the ratio γ_∞ between the reference heat capacities:

$$M = \frac{U_0}{\sqrt{\gamma_\infty \frac{R}{W_\infty} T_\infty}}, \quad \text{with} \quad \gamma_\infty = \frac{C_{p,\infty}}{C_{v,\infty}}. \tag{3.14}$$

3.3 Kinetic spray modeling

In this section, we consider a dilute disperse phase $10^{-4} < \alpha_l < 10^{-2}$, which consists of small droplets supposed of spherical shape. In this case, the gas-liquid interface is not solved. This assumption is valid for a low aerodynamic Weber number, which for a droplet of diameter d is defined as follows:

$$\mathrm{We}_g = \frac{\rho_g d ||\boldsymbol{u}_g - \boldsymbol{u}_l||^2}{\sigma}. \tag{3.15}$$

The Weber number characterizes the ratio of the inertial forces with the surface tension effects, as estimated by the surface tension σ. A critical Weber number $\mathrm{We}_{cri} \simeq 12$ is often used to classify the droplets into stable and unstable. In the disperse phase region of the fuel injection, the Weber number of the droplets obtained after the primary breakup, can vary from low values $\mathrm{We}_g < \mathrm{We}_{cri}$ that corespondents to very small and stable droplets, and medium or high values $\mathrm{We}_g >> \mathrm{We}_{cri}$, for which the droplets can undergo secondary breakup to generate smaller droplets of low Weber number.

The so-called kinetic approach represents the spray at a mesoscopic level by using a number density function (NDF). In this framework, we use a statistical description of the droplets population based on a set of variables characterizing the droplets (size, velocity, temperature, etc.). These variables are used as the internal variables of the NDF. Thus, the NDF can give rich information about the spray such as the polydispersion and the polykinetic (velocity distribution) spray features. In the following, we denote by $\widehat{\boldsymbol{\xi}}_d$ to the internal variables of the NDF. We use the ensemble average over realizations (see chapter 2) to define the NDF. Considering N_{max}, the maximum droplet number in the domain, we define the NDF as follows:

$$f(t, \boldsymbol{x}; \widehat{\xi}_d) = \left\langle \sum_{j=1}^{N_{max}} \delta(\boldsymbol{x} - \boldsymbol{x}_j(t)) \delta(\widehat{\xi}_d - \xi_{d,j}(t)) \right\rangle, \tag{3.16}$$

where $\langle \bullet \rangle$ is the ensemble average operator defined in section 2.10 and $\boldsymbol{x}_j(t)$ and $\xi_{d,j}(t)$ are respectively the position and the phase variable of droplet j at time t for one realization.

3.3.1 William-Boltzmann equation

In spray modeling for combustion, we often characterize each droplet, by its size ϕ, its velocity $\boldsymbol{c}$, and its temperature T. These three variables are going to be the internal variables $\boldsymbol{\xi}_d = (\phi, \boldsymbol{c}, T_d)$ of the NDF. The size variable can be the volume, the surface or the radius of the droplet. We can also consider more than one size variable to characterize the droplets, when the droplet shape is not spherical. In the following, we suppose a spherical shape for droplets and we use the surface area as the only size variable $\phi = S$. The number density function $f(t, \boldsymbol{x}, S, \boldsymbol{c}, T_d)$ represents the probable density number of droplets located at position $\boldsymbol{x}$, traveling with velocity $\boldsymbol{c}$ and having size S and temperature T_d. To derive a transport equation for the NDF, we further assume droplet independence, that can be linked with the molecular chaos assumption in the kinetic theory of gases. We can then derive the following Generalized Population Balance equation

$$\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{c} f) + \partial_{\boldsymbol{c}} \cdot (\boldsymbol{a} f) + \partial_S (\mathrm{R}_S f) + \partial_{T_d} (\mathrm{E}_T f) = \Gamma + \mathrm{Q}, \tag{3.17}$$

known also by the Williams-Boltzmann (WB) equation Williams(1958), where:

- $\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{c} f)$ represents the free transport of the spray;
- $\boldsymbol{a} = \mathrm{d}_t(\boldsymbol{c})$ is the acceleration applied on the droplets per unit mass;
- $\mathrm{R}_S = \mathrm{d}_t(S)$ is the rate of change of the size S of the droplets;
- $\mathrm{E}_T = \mathrm{d}_t(T_d)$ is the rate of change of the droplets temperature due to heat transfer;
- Γ is the rate of change of distribution function f due to collisions;
- Q is the rate of change of f due to particles formation by secondary break-up.

3.3.2 Source term closure models

Acceleration the droplet acceleration is due to different forces applied on it. In general, we can decompose the forces into four contributions:

$$\boldsymbol{a} = \frac{1}{m_p}(\boldsymbol{F}_G + \boldsymbol{F}_D + \boldsymbol{F}_{vm} + \boldsymbol{F}_L), \tag{3.18}$$

where m_p is the mass of the droplet, $\boldsymbol{F}_G$ is the buoyancy force, $\boldsymbol{F}_D$ is the drag force applied by the gas, $\boldsymbol{F}_{vm}$ is the virtual mass force and corresponds to the acceleration of the gas carried by the particle and $\boldsymbol{F}_L$ represents the lift force due to droplet rotation.

In the following, we focus on the drag term. Indeed, it was shown in Dufour(2005) that for a gas-liquid flow where $\rho_g/\rho_l < 10^{-2}$, the drag force is the dominant force. A general expression for the drag force is given by, O'Rourke(1981); Sirignano(1999).

$$\boldsymbol{F} = \frac{1}{8}\,\rho_g\, C_D\, S\, \left(\boldsymbol{u}_g - \boldsymbol{c}\right) ||\boldsymbol{u}_g - \boldsymbol{c}||, \tag{3.19}$$

where C_D is the drag coefficient. This coefficient depends on the particle shape, as well as on the flow parameters such as the Reynolds number, the Mach number, the turbulence, etc. For spherical droplets, we can focus only on the variation of C_D with the Reynolds number based on the relative velocity:

$$\mathrm{Re}_p = \frac{\rho_g S^{1/2} ||\boldsymbol{u}_g - \boldsymbol{c}||}{\sqrt{\pi}\mu_g}. \tag{3.20}$$

Two regimes can be considered: first the Stokes flow regime where the drag coefficient varies inversely with the Reynolds number, for $\mathrm{Re}_p < 1000$; second the inertial range regime where the drag coefficient approaches a nearly constant value, see Clift et al.(1978) for details. For low relative velocity, the drag coefficient is given by the Stokes law, proposed by C.G. Stokes in 1851:

$$C_D = \frac{24}{\mathrm{Re}_p}, \quad \mathrm{Re}_p < 1. \tag{3.21}$$

In this case the drag force term writes:

$$\boldsymbol{a} = m_p \frac{18\,\pi\,\mu_g}{\rho_l\,S}\left(\boldsymbol{u}_g - \boldsymbol{c}\right) = m_p \frac{(\boldsymbol{u}_g - \boldsymbol{c})}{\tau_p(S)}, \tag{3.22}$$

where

$$\tau_p(S) = \frac{\rho_l\,S}{18\,\pi\,\mu_g}, \tag{3.23}$$

is a characteristic response time of the droplet to the gas dynamic. for the sake of simplicity, the Stokes law is used to model the force acting on the droplets in the following.

Evaporation and heat rate the droplet size decreases due to the evaporation. Its dilatation due to heating is neglected in this study. We assume that the evaporation process is slow enough to consider a steady state and the evaporated liquid is diffused in the gaseous phase following a Fick's law. Thus, in the limit of no chemical reaction, R_S reads Sirignano(1999):

$$\mathrm{R}_S = 4\pi \frac{\rho_g}{\rho_l}\,\mathrm{Sh_c} D_{yf} \log(1 + B_m), \tag{3.24}$$

where B_m is the spalding mass transfer number, $\mathrm{Sh_c}$ is the convective Sherwood number and D_{yf} is the Fick's law binary coefficient. Further details about these quantities are given in de Chaisemartin(2009). If no other convective correction is taken into account, it leads to the d^2 law, Godsave(1953); Spalding(1953):

$$\mathrm{R}_S = -K, \tag{3.25}$$

where K is a constant independent of the droplet size, but which may depend on the local gas condition. In the following, we adopt this model to close the evaporation rate.

A model for the heat rate can be also derived from the film theory around an isolated droplet. In Sirignano(1999), the heat rate is expressed, for the same condition considered to express the evaporation rate, as follows:

$$\mathrm{E}_T = 6\pi \frac{\rho_g}{\rho_l S C_{p,l}} \mathrm{Sh_c} D_{yf} \log(1 + B_m)\left(\frac{C_{p,g}(T_g(\infty) - [T]_s}{B_T} - L_v\right), \tag{3.26}$$

where $T_g(\infty)$ is the gas temperature beyond the diffusive thermal boundary layer, $[T]_s$ is the temperature at the droplet surface, L_v is the latent heat of vaporization and B_T is the spalding dimensionless heat transfer number. Let us give expressions for spalding dimensionless transfer numbers and the relation between them:

$$B_m = \frac{[Y_F]_s - [Y_F]_\infty}{1 - [Y_F]_s}, \quad B_T = \frac{\rho_g ||\boldsymbol{u}_g - \boldsymbol{c}|| C_{p,g}}{h_c}, \quad B_T = (B_m + 1)^{\frac{\mathrm{Sh_c}}{\mathrm{Nu_c}} \frac{\mathrm{Sh}}{\mathrm{Pr}}}, \tag{3.27}$$

where $[Y_F]_s$ is the fuel mass fraction at the droplet surface, $[Y_F]_\infty$ is the fuel mass fraction beyond the diffusive mass boundary layer and h_c is the convective modified heat transfer coefficient. According to the film theory, the relative velocity between the gas and the droplet reduces boundary layer around the droplets. Therefore some correction models on the Sherwood and the Nusselt numbers have been proposed in the literature Abramzon and Sirignano(1989); Sirignano(1999); Faeth(1983).

Coalescence and collision source terms The velocity dispersion of droplets around the mean motion can cause collisions between droplets. Different possible scenarios can be considered when two or more droplets collide. The droplets may coalesce to form one single droplet, rebound or explode to form a large number of other droplets. In the following, we consider collision kinetic modeling for spray, which is detailed in Drew and Passman(1999). The spray is assumed to be dilute. Therefore, binary collisions (collision of two droplets) are the most probable scenario. The collision source term in WB equation has two contributions $\Gamma = \Gamma_c + \Gamma_r$, where Γ_c represents collisions resulting in coalescence and Γ_r collisions resulting in rebounds.

As presented in Laurent(2002); Laurent et al.(2004); Doisneau et al.(2013), the coalescence term is written as the sum of the coalescence part Γ_c^+ forming droplets at size and velocity $(S, \boldsymbol{c})$ and the coalescence part Γ_c^- which leads to the disappearance of the droplets parametrized by $(S, \boldsymbol{c})$. Following the same derivation as in Laurent(2002), we can express the two terms as follows:

$$\begin{aligned} \Gamma_c^+ &= \frac{1}{2} \int_{S^\star} \int_{\boldsymbol{c}^\star} E_{\mathrm{coal}}(S, S^\star) B(S^\diamond, S^\star) |\boldsymbol{c}^\diamond - \boldsymbol{c}^\star| f(S^\diamond, \boldsymbol{c}^\diamond) f(S^\star, \boldsymbol{c}^\star) J \,\mathrm{d}S^\star \mathrm{d}\boldsymbol{c}^\star, \\ \Gamma_c^- &= -\int_{S^\star} \int_{\boldsymbol{c}^\star} E_{\mathrm{coal}}(S, S^\star) B(S, S^\star) |\boldsymbol{c} - \boldsymbol{c}^\star| f(S, \boldsymbol{c}) f(S^\star, \boldsymbol{c}^\star) J \mathrm{d}S^\star \mathrm{d}\boldsymbol{c}^\star. \end{aligned} \tag{3.28}$$

where E_{coal} is the probability of the coalescence of two colliding droplets. In the production term, we count all the possible collisions between a droplet parametrized by $(S^\star, \boldsymbol{c}^\star)$ and a droplet parametrized $(S^\diamond, \boldsymbol{c}^\diamond)$ resulting in a single droplet parametrized by $(S, \boldsymbol{c})$. For fixed $(S^\star, \boldsymbol{c}^\star)$, the mass and momentum conservation leads to a one-to-one transformation $(S, \boldsymbol{c}) \rightarrow (S^\diamond, \boldsymbol{c}^\diamond)$:

$$\begin{aligned} S^{\diamond 3/2} &= S^{3/2} - S^{\star 3/2}, \\ \boldsymbol{c}^\diamond &= \frac{S^{3/2}\boldsymbol{c} - S^{\star 3/2}\boldsymbol{c}^\star}{S^{3/2} - S^{\star 3/2}}. \end{aligned} \tag{3.29}$$

J is the Jacobian of this transformation. The collision cross section is:

$$B(S, S^\star) = \pi \left(R(S) + R(S^\star) \right)^2, \tag{3.30}$$

where the droplet radius is $R(S) = (S/(4\pi))^{1/2}$.

The rebound term Γ_r can be modeled using the generalized Boltzmann collision operator for hard-sphere inelastic binary collisions Drew and Passman(1999):

$$\begin{aligned} \Gamma_r = & \int_0^\infty (1 - E_{\mathrm{coal}}(S, S^\star)) B(S, S^\star) \\ & \frac{1}{\pi} \int_{R^3} \int_{S^+} \left[\frac{f(S, \boldsymbol{c}'') f(S^\star, \boldsymbol{c}'')}{\alpha(S, S^\star)^2} - f(S, \boldsymbol{c}) f(S^\star, \boldsymbol{c}^\star) \right] |\boldsymbol{g} \cdot \boldsymbol{n}| \,\mathrm{d}\boldsymbol{n} \mathrm{d}\boldsymbol{c}^\star \mathrm{d}S^\star, \end{aligned} \tag{3.31}$$

where $\boldsymbol{n} = (\boldsymbol{x} - \boldsymbol{x}^*)/|\boldsymbol{x} - \boldsymbol{x}^*|$ is the unit vector in the direction between the droplet centers, $\boldsymbol{g} = \boldsymbol{c} - \boldsymbol{c}^\star$ is the relative velocity before collision, and $0 \leq \alpha \leq 1$ is the coefficient of restitution with the property $\alpha(S, S^\star) = \alpha(S^\star, S)$. The surface S^+ is the unit half sphere on which $\boldsymbol{g} \cdot \boldsymbol{n} > 0$. The double-prime variables denote values before the inverse collision, which are defined in terms of the pre-collision values by Fox and Vedula(2010).

$$
\begin{aligned}
\boldsymbol{c}'' &= \boldsymbol{c} - \frac{S^{\star 3/2}(1+\alpha(S,S^\star))}{\alpha(S,S^\star)(S^{3/2}+S^{\star 3/2})}(\boldsymbol{g}\cdot\boldsymbol{n})\boldsymbol{n}, \\
\boldsymbol{c}^{\star\prime\prime} &= \boldsymbol{c}^\star + \frac{S^{3/2}(1+\alpha(S,S^\star))}{\alpha(S,S^\star)(S^{3/2}+S^{\star 3/2})}(\boldsymbol{g}\cdot\boldsymbol{n})\boldsymbol{n}.
\end{aligned}
\tag{3.32}
$$

In the case of elastic rebounds, one has $\alpha = 1$.

Breakup source terms the secondary breakup of droplets is mainly due to aerodynamic forces of the gas, where high relative velocity between the gas and the disperse phase causes the deformation and eventually breakage of the droplets. The surface tension of the gas-liquid interface counteracts this process. Thus, the dimensionless Weber number is a physical criterion this phenomenon:

$$
\mathrm{We} = \frac{2R\rho_g||\boldsymbol{u}_g - \boldsymbol{c}||}{\sigma},
\tag{3.33}
$$

where R is the droplet radius, $\boldsymbol{c}$ its velocity and σ is the surface tension. For Weber number $\mathrm{We} > 12$, secondary break-up has to be taken into account. In this part, we recall briefly the main points to model the break-up operator Q, where more details can be found in Hylkema(1999); Villedieu and Hylkema(2000).

The breakup operator can be decomposed into two contributions $\mathrm{Q} = \mathrm{Q}^+ + \mathrm{Q}^-$:

$$
\begin{aligned}
\mathrm{Q}^+(t,\boldsymbol{x};\boldsymbol{c},S) &= \int_{\boldsymbol{c}^\star}\int_{S^\star} \nu^{bup}(\boldsymbol{c}^\star,S^\star)h(\boldsymbol{c},S,\boldsymbol{c}^\star,S^\star)f(t,\boldsymbol{x},\boldsymbol{c}^\star,S^\star)dS^\star d\boldsymbol{c}^\star, \\
\mathrm{Q}^-(t,\boldsymbol{x};\boldsymbol{c},S) &= -\nu^{bup}(\boldsymbol{c},S)f(t,\boldsymbol{x},\boldsymbol{c},S),
\end{aligned}
\tag{3.34}
$$

where $h(\boldsymbol{c},S,\boldsymbol{c}^\star,S^\star)$ is the number of droplets parametrized by $(\boldsymbol{c},S)$ produced by the breakage of a droplet parametrized by $(\boldsymbol{c}^\star,S^\star)$ and $\nu^{bup}(\boldsymbol{c},S)$ is an average frequency of breakup for a droplet parameterized by $(\boldsymbol{c},S)$. These two functions are often expressed as function of the local gas properties and the Weber number, using empirical relations, Dufour(2005); O'Rourke and Amsden(1987).

3.3.3 Gas-spray source terms

The source terms arising in the gaseous equations (3.1) and which are related to the influence of the dispersed liquid phase on the carrier gaseous phase can be now expressed thanks to the kinetic spray modeling. These terms represent:

- the variation of mass density, for the mixture and for individual species, due to spray vaporization,
- the variation of momentum of the spray, due to spray vaporization and to external forces,
- and finally the variation of enthalpy per unit mass, due to spray vaporization and to heat transfer.

$$
\begin{aligned}
S^{m} &= -\iiint \rho_l \, \frac{\sqrt{S}}{4\sqrt{\pi}} \, \mathrm{R}_S f \mathrm{d}S \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}T, \\
S_k^{species} &= -\iiint \Omega_{\mathrm{k}}^{\mathrm{Surf}} \, \rho_l \, \frac{\sqrt{S}}{4\sqrt{\pi}} \, \mathrm{R}_S \, f \mathrm{d}S \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}T, \\
S^{mom} &= -\iiint \rho_l \, \boldsymbol{c} \, \frac{\sqrt{S}}{4\sqrt{\pi}} \, \mathrm{R}_S \, f \mathrm{d}S \, \mathrm{d}_{,}\mathrm{d}T - \iiint \rho_l \, \frac{S^{3/2}}{6\sqrt{\pi}} \, \boldsymbol{F} \, f \, \mathrm{d}S \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}T, \\
S^{enth} &= -\iiint \rho_l \, \mathrm{C}_{p,l} \, T \, \frac{\sqrt{S}}{4\sqrt{\pi}} \, \mathrm{R}_S \, f \mathrm{d}S \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}T - \iiint \rho_l \, \frac{S^{3/2}}{6\sqrt{\pi}} \, \mathrm{C}_{p,l} \, \mathrm{E}_T \, f \, \mathrm{d}S \, \mathrm{d}\boldsymbol{c} \, \mathrm{d}T,
\end{aligned}
\tag{3.35}
$$

where $\Omega_{\mathrm{k}}^{\mathrm{Surf}}$ is the evaporation flux fraction for species k, from liquid surface to gas. If we suppose that all fuel species evaporate in the same way, the evaporation flux fraction of the species is given by the mass fraction of the species in the liquid fuel decomposition: $\Omega_{\mathrm{k}}^{\mathrm{Surf}} = \rho_{k,l}/\rho_l$, where $\rho_{k,l}$ is the mass concentration of the species k in liquid fuel. One has to note that to obtain the expression S^{enth}, we assume that no reaction is happening at the droplet surface. We thus consider that droplets evaporate before burning.

Two-way coupling is taken into account here, the gas influence on the spray resulting in $\boldsymbol{F}$, R_S and E_T coefficients, depending on both gas and liquid local properties. The influence of the spray over the gas is taken into account by the source terms of Eq. (3.35).

3.3.4 Simplified modeling framework

In the following of this chapter, we place ourselves in the region beyond a secondary break-up. After this process, the droplets are stable since the cohesive forces due to the surface tension are stronger than the destructive forces related mainly to the aerodynamic forces. We can then neglect the break-up operator. The thermal inertia can also be neglected when the droplets are small enough. Finally, we consider a dilute spray, where the collision between the droplets is also negligeable. Therefore, the NDF becomes $f(t, \boldsymbol{x}, \boldsymbol{c}, S)$ and the WB equation (3.17) is reduced to:

$$
\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{c} f) + \partial_{\boldsymbol{c}} \cdot (\boldsymbol{F} f) + \partial_S \, (\mathrm{K} f) = 0, \tag{3.36}
$$

where $\boldsymbol{F} = \dfrac{\boldsymbol{u}_g - \boldsymbol{c}}{\tau_p(S)}$ is the drag force modeled by using the Stokes law as it was proposed in (3.22) and K is a constant evaporation rate given by the d^2 evaporation law, (3.25).

3.3.5 Dimensionless William-Boltzmann equation

In this section, we derive a dimensionless kinetic equation. We consider the reference gas variables defined in section 3.2.3 and some additional variables that characterize the droplets: the reference size S_0 and the reference droplet density $n_0 = N_0/x_0^d$, for a physical space with d spatial dimensions and total droplet number in the entire domain N_0. We introduce the dimensionless variables of the disperse phase:

$$
\begin{aligned}
\overline{S} &= \frac{S}{S_0}, \\
\overline{\boldsymbol{c}} &= \frac{\boldsymbol{c}}{U_0}.
\end{aligned}
\tag{3.37}
$$

In this framework, the dimensionless NDF $\overline{f}$ can be written as follows:

$$\overline{f}(\overline{t}, \overline{\boldsymbol{x}}, \overline{S}, \overline{\boldsymbol{c}}) = \frac{U_0^d S_0}{n_0} f(t, \boldsymbol{x}, S, \boldsymbol{c}). \tag{3.38}$$

The dimensionless formulation for the drag force and the evaporation rate is given by:

$$\begin{aligned} \overline{\boldsymbol{F}} &= \frac{x_0}{U_0^2} \boldsymbol{F}, \\ \overline{\mathrm{K}} &= \frac{x_0}{U_0 S_0} \mathrm{K}. \end{aligned} \tag{3.39}$$

The dimensionless drag force can also be rewritten as follows:

$$\overline{\boldsymbol{F}} = \frac{\overline{\boldsymbol{u}_g} - \overline{\boldsymbol{c}}}{\mathrm{St}(\overline{S})} \tag{3.40}$$

where $\mathrm{St}(S) = S_0 \tau_p(S)/\tau_g$ is the Stokes number, which characterizes the response of the droplet to the gas dynamics. Considering dimensionless variables, William-Boltzmann equation is rewritten as follows:

$$\partial_{\overline{t}} \overline{f} + \partial_{\overline{\boldsymbol{x}}} \cdot \left(\overline{\boldsymbol{c}} \overline{f} \right) + \partial_{\overline{\boldsymbol{c}}} \cdot \left(\frac{\overline{\boldsymbol{u}_g} - \overline{\boldsymbol{c}}}{\mathrm{St}(\overline{S})} \overline{f} \right) - \overline{\mathrm{K}}\ \partial_{\overline{S}} \overline{f}. = 0, \tag{3.41}$$

In the following of this chapter, we only use dimensionless quantities, and for the sake of legibility, we will omit the bar sign. We also use a reference size equal to the maximum droplet size, such that $S \in [0, 1]$.

3.4 Eulerian moment method

Given the high dimension of the variables space of the WB equation, its discretization for industrial applications is not attainable under reasonable CPU time. Since a high accuracy on the resolution of the distribution of droplets is not necessary for the applications we are concerned with, and only macroscopic quantities are needed, an Eulerian moment method can be a promising alternative for an approximate resolution. In this approach, we consider the moments of the NDF over the phase space (velocity and size):

$$M_{i,j,k,l} = \int_0^1 \int_{\mathbb{R}^3} S^l c_x^i c_y^j c_z^k f(t, \boldsymbol{x}, \boldsymbol{c}, S) dS d^3\boldsymbol{c}. \tag{3.42}$$

The moments of the NDF are macroscopic features of the statistics of the flow. As we increase the number of moments, we approach further to the mesoscopic description given by the NDF. From the WB equation (3.36), we can derive transport equations on the moments of the NDF in the physical space. With this method, we reduce the dimension to the time and space dimension, where numerical resolution is affordable for realistic configurations. However, reducing the dimension of the phase space variables will generate some modeling difficulties. Indeed, the use of moments method results in information loss, and yields an unclosed system of moments. In fact, the transport equation of a given set of size-velocity moments involves other moments of higher order or, in some cases like evaporation, the knowledge of the entire size distribution Massot et al.(2010). Therefore, we need closure assumption for the unclosed terms. The closure assumptions depend on the set of transported moments. For some applications, we need to transport a large number of moments in velocity and size in order to be accurate enough. To choose an adequate modeling strategy, two main spray features should be taken into account:

- Polykinetic feature of the droplet velocity distribution, i.e. the co-existence of several droplet velocities at the same time and location, is a peculiarity of the disperse phase dynamics. Due to the difference in droplet inertia, the droplet velocities can be uncorrelated, which can lead to different droplet velocities at the same location. This issue can be neglected in the case of very small droplets, characterized by a low Stokes number, where the droplets are more likely to follow the gas flow. But in the case of medium and large droplets, we can encounter Particle Trajectory Crossings (PTC). The PTC can occur for droplets of different sizes (hetero-PTC) or of the same size (homo-PTC). In this case, one needs to keep some information about the velocity dispersion in the Eulerian framework by considering higher order velocity moments.
- Polydisperse feature of the droplet size distribution, i.e. the co-existence of droplets of different sizes at the same location, is an important parameter to consider in sprays modeling. Indeed, the different interactions with the continuous phase depend on the droplet size, such that the response of two droplets of different sizes to the continuous phase is different. Furthermore, the polydispersion has a direct impact on the polykinetic feature, since the droplets velocity is highly correlated to their size distribution, Vié et al.(2013b).

In the following, we discuss these two important issues, which are related to moment modeling and we present some existing modeling strategy to cope with these Eulerian modeling difficulties. In the following, we consider that the NDF has the following presumed form:

$$f(t,\boldsymbol{x};S,\boldsymbol{c}) = n(t,\boldsymbol{x},S)\phi_{\boldsymbol{u}}(t,\boldsymbol{x},\boldsymbol{c}-\boldsymbol{u}_d(t,\boldsymbol{x},S)), \tag{3.43}$$

where $\int_{\boldsymbol{c}} \phi_{\boldsymbol{u}}(t,\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c} = 1$ and $\int_{\boldsymbol{c}} \boldsymbol{c}\phi_{\boldsymbol{u}}(t,\boldsymbol{x},\boldsymbol{c})d\boldsymbol{c} = 0$.

3.4.1 Eulerian Polykinetic modeling

In this part, we discuss some models aimed at capturing the velocity distribution: starting from the basic mono-kinetic assumption for the velocity distribution, where we consider a unique spray velocity for all droplets at a given time and position, to some more complex models where we take into account the variance in the velocity distribution and some anisotropy features of jet crossings. The main difficulty for dilute spray consists in dealing with high Knudsen flow regimes. In such case, we can not use some equilibrium assumption as in the kinetic theory for dense gas flows, where the collision operator in the Boltzmann equation leads to a Gaussian velocity distribution. Thus, in dilute spray, as for rarefied gas, the velocity distribution has no preferential form. In the literature, two approaches are used to close the velocity moments transport equations: 1- Algebraic-ClosureBased Moment Methods (ACBMM), where a limited set of moments are transported, usually up to second order and the higher order moments involved in the system are expressed explicitly using lower moments thanks to an equilibrium assumption inspired from RANS turbulence modeling. 2- Kinetic-Based Moment Methods (KBMM), which we use in the following to model the polykinetic aspect of the spray. In this approach, we transport a set of moments of the velocity distribution. Then, in order to close at higher order moments involved in the transport equation, we associate the transported moments to a family of velocity distributions with a sufficient number of parameters to control the given set of moments. Two possible forms of the distribution closures have been studied in the literature. The first one consists in using quadrature methods, where the distribution is approximated by a sum of weighted Dirac distributions see Marchisio et al.(2003) and references therein. The second approach consists in using an anisotropic Gaussian distribution Vié et al.(2015a); Dupif(2018) ,or multi-Gaussian distribution, when moments up to the second order are used. The anisotropic

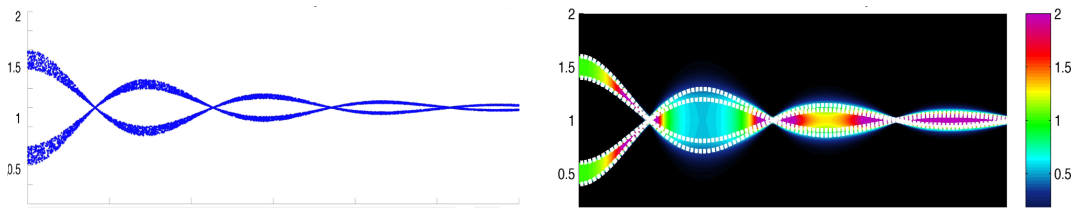


Figure 3.2: *Steady solution of two inertial particle jets ($St = 20$) injected in a compressive velocity field:1- In left Lagrangian particles 2-In right number density (m^{-3}) solution obtained with Anisotropic Gaussian model Doisneau(2013) and white lines represent the lower and upper Lagrangian trajectories for each jet.*

Gaussian has robust mathematical properties (hyperbolicty) and shows a sufficient capacity in capture PTC as illustrated in Figure 3.2.

In this paragraph, we present briefly the model that we can derive, when assuming anisotropic Gaussian distribution in velocity, we also suppose that the size and the velocity are uncorrelated: $\boldsymbol{u}_d(t,\boldsymbol{x};S) = \boldsymbol{u}_d(t,\boldsymbol{x})$. The velocity distribution expresses as follows:

$$\phi_{\boldsymbol{u}}(t,\boldsymbol{x},\boldsymbol{c}) = \frac{1}{(2\pi)^{d/2}\det\boldsymbol{\Sigma}}\exp\left(-\frac{1}{2}\boldsymbol{c}^T\boldsymbol{\Sigma}^{-1}\boldsymbol{c}\right), \tag{3.44}$$

where d is the spatial dimension and $\Sigma = \frac{1}{n}\int_c (\boldsymbol{c}-\boldsymbol{u}_p)\otimes(\boldsymbol{c}-\boldsymbol{u}_p) f \; d\boldsymbol{c}$ is the covariance matrix.

Integrating equation (3.36), multiplied by the vector $(1,\boldsymbol{c},\boldsymbol{c}\otimes\boldsymbol{c})^t$, over the velocity phase space leads to the following system of conservation laws with source terms related to the drag acceleration:

$$\partial_t \mathbf{m} + \nabla\cdot(\mathcal{F}(\mathbf{m})) \quad = \quad S(\mathbf{m}). \tag{3.45}$$

In two dimensions ($d = 2$), we transport six moments, which corresponds to the number density $n(t,\boldsymbol{x})$, the mean vector $\boldsymbol{u} = (u,v)^T$ and the superior triangular part of the symmetric energy tensor $n\boldsymbol{E} = \frac{1}{2}n(\boldsymbol{u}\otimes\boldsymbol{u}+\boldsymbol{\Sigma})$:

$$\mathbf{m} = \begin{pmatrix} m_{0,0} \\ m_{1,0} \\ m_{0,1} \\ m_{2,0} \\ m_{1,1} \\ m_{0,2} \end{pmatrix} = n\begin{pmatrix} 1 \\ u \\ v \\ u^2+\Sigma_{11} \\ uv+\Sigma_{12} \\ v^2+\Sigma_{22} \end{pmatrix}. \tag{3.46}$$

The flux vector $\mathcal{F}(\mathbf{m})$ depends on the third order velocity moments. Under the assumption on the shape of the velocity distributions (3.44), those can be expressed as function of the covariance matrix $\boldsymbol{\Sigma}$ and of the mean velocity vector $\boldsymbol{u}_d = (u,v)$:

$$\mathcal{F}(\mathbf{m}) = \begin{pmatrix} u & v \\ u^2+\Sigma_{11} & uv+\Sigma_{12} \\ vu+\Sigma_{12} & v^2+\Sigma_{22} \\ u^3+3u\Sigma_{11} & u^2v+v\Sigma_{11}+2u\Sigma_{12} \\ u^2v+2u\Sigma_{12}+v\Sigma_{11} & uv^2+2v\Sigma_{12}+u\Sigma_{22} \\ uv^2+u\Sigma_{22}+2v\Sigma_{12} & v^3+3v\Sigma_{22} \end{pmatrix}. \tag{3.47}$$

Finally, the source term corresponding to the drag effects following a Stokes law reads:

$$S(\mathbf{m}) = \frac{n}{\mathrm{St}(S)} \begin{pmatrix} 0 \\ u_g - u \\ v_g - v \\ 2uu_g - 2u^2 - 2\Sigma_{11} \\ uv_g + u_g v - 2uv - 2\Sigma_{12} \\ 2vv_g - 2v^2 - 2\Sigma_{22} \end{pmatrix}. \tag{3.48}$$

The LHS of the system can be written as the following form:

$$\begin{array}{rcl} \partial_t n + \nabla \cdot (n\boldsymbol{u}_d) & = & 0, \\ \partial_t n\boldsymbol{u}_d + \nabla \cdot (n\boldsymbol{u}_d \otimes \boldsymbol{u}_d + \boldsymbol{P}) & = & 0, \\ \partial_t n\boldsymbol{E} + \nabla \cdot ((n\boldsymbol{E} + \boldsymbol{P})\boldsymbol{u}_d) & = & 0, \end{array} \tag{3.49}$$

where the pressure tensor $\boldsymbol{P}$ is related to the covariance matrix $\boldsymbol{\Sigma}$ as follows:

$$\boldsymbol{P} = n\boldsymbol{\Sigma}. \tag{3.50}$$

The covariance matrix $\boldsymbol{\Sigma}$ which measures the velocity dispersion in different space directions can be expressed algebraically as a function of the moments up to second order:

$$\Sigma_{11} = \frac{m_{2,0} - m_{1,0}m_{1,0}}{m_{0,0}}, \quad \Sigma_{12} = \frac{m_{1,1} - m_{1,0}m_{0,1}}{m_{0,0}}, \quad \Sigma_{22} = \frac{m_{0,2} - m_{0,1}m_{0,1}}{m_{0,0}}. \tag{3.51}$$

This model degenerates to the Gaussian isotropic model, when the covariance matrix is proportional to the identity matrix $\boldsymbol{\Sigma} = \sigma I_d$. In this case, it is sufficient to take only the trace of the energy tensor $ne = n\mathrm{Tr}(\boldsymbol{E}) = \int_{\boldsymbol{c}} ||\boldsymbol{c}||^2 f(\boldsymbol{c}) d\boldsymbol{c}$ into account. The system of equations (3.49) becomes:

$$\begin{array}{rcl} \partial_t n + \nabla \cdot (n\boldsymbol{u}_d) & = & 0, \\ \partial_t n\boldsymbol{u}_d + \nabla \cdot (n\boldsymbol{u}_d \otimes \boldsymbol{u}_d + p) & = & 0, \\ \partial_t ne + \nabla \cdot ((ne + p)\boldsymbol{u}_d) & = & 0, \end{array} \tag{3.52}$$

where $p = n\sigma = n(e - \frac{1}{2}||\boldsymbol{u}_d||^2)/d$ and d is again the space dimension. The two last systems of equations are strictly hyperbolic. In the case when the velocity dispersion is neglected $\sigma = 0$, the model degenerates to the pressureless model:

$$\begin{array}{rcl} \partial_t n + \nabla \cdot (n\boldsymbol{u}_d) & = & 0, \\ \partial_t n\boldsymbol{u}_d + \nabla \cdot (n\boldsymbol{u}_d \otimes \boldsymbol{u}_d) & = & 0, \end{array} \tag{3.53}$$

This last model corresponds to the monokinetic assumption and which is only weakly hyperbolic: the system of equation in each transport direction has one real eigenvalue given by the velocity. The monokinetic assumption is valid for low Stokes numbers, where PTC can be neglected since the particles acts as tracers of the gas field. In the case where PTC occurs, this model leads to singular accumulation of droplets, and so to an overestimation of droplet segregation compared with anisotropic and isotropic Gaussian models Vié et al.(2015a); Sabat(2016). Another way to capture PTC consists in using quadrature methods like QMOM method, see Marchisio and Fox(2005) for more details.

3.4.2 Polydisperse sprays

3.4.2.1 Multi-fluid approach

Multi-fluid approach or sectional method consists in discretizing the size variables into compact intervals, called *size-sections*. The discretization in size is given by a finite sequence $(S_i)_{1\leq i\leq N_s}$, where $N_s > 1$ is the number of size-sections and we have:

$$0 = S_0 < S_2 < \ldots < S_{N_s} = 1. \tag{3.54}$$

A system of conservation laws can be then derived for averaged variables over each size-section. These averaged quantities correspond to moments of the NDF integrated on each interval $[S_j, S_{j+1}]$. In this part, we present a first order version of the multi-fluid approach Laurent et al.(2004); de Chaisemartin(2009), where only one size and one velocity moments are used for each size-section, this approach is called the One-Size Moment (OSM) method. In the OSM method, the size and the velocity distributions are two piecewise constant functions of the size, such that for $S \in [S_j, S_{j+1}]$ we have:

$$\begin{aligned} n(t,\boldsymbol{x},S) &= \rho^{(j)}\kappa^{(j)}, \\ \boldsymbol{u}_d(t,\boldsymbol{x},S) &= \boldsymbol{u}_d^{(j)}(t,\boldsymbol{x}), \end{aligned} \tag{3.55}$$

where $\rho^{(j)}$ is the mass density of the droplets in the j-th section, given by:

$$\rho^{(j)}(t,\boldsymbol{x}) = \int_{S_j}^{S_{j+1}} \frac{\rho_l}{6\sqrt{\pi}} S^{3/2} n(S)dS, \tag{3.56}$$

so, that:

$$\kappa^{(j)} = \frac{6\sqrt{\pi}}{\rho_l}\frac{5/2}{S_{j+1}^{5/2} - S_j^{5/2}}. \tag{3.57}$$

Then, we can derive conservation equations for each size-section, by integrating the semi kinetic system (3.53) and by assuming a monokinetic-assumption in each section:

$$\begin{aligned} \partial_t \rho^{(j)} + \nabla\cdot\left(\rho^{(j)}\boldsymbol{u}_d^{(j)}\right) &= -\left(E_1^{(j)} + E_2^{(j)}\right)\rho^{(j)} + E_1^{(j+1)}\rho^{(j+1)}, \\ \partial_t \rho^{(j)}\boldsymbol{u}_d^{(j)} + \nabla\cdot\left(\rho^{(j)}\boldsymbol{u}_d^{(j)}\otimes\boldsymbol{u}_d^{(j)}\right) &= -\left(E_1^{(j)} + E_2^{(j)}\right)\rho^{(j)}\boldsymbol{u}_d^{(j)} \\ &\quad + E_1^{(j+1)}\rho^{(j+1)}\boldsymbol{u}_d^{(j+1)} + \rho^{(j)}\boldsymbol{F}^{(j)}. \end{aligned} \tag{3.58}$$

where the averaged drag force in section j is:

$$\begin{aligned} \rho^{(j)}\boldsymbol{F}^{(j)} &= \int_{S_j}^{S_{j+1}} \frac{\rho_l}{6\sqrt{\pi}} S^{3/2}\frac{\boldsymbol{u}_g(t,\boldsymbol{x}) - \boldsymbol{u}_d(t,\boldsymbol{x},S)}{\mathrm{St}(S)} n(t,\boldsymbol{x},S)dS, \\ &= \rho^{(j)}\frac{5}{3}\frac{(S_{j+1}^{3/2} - S_j^{3/2})\overline{S}}{(S_{j+1}^{5/2} - S_j^{5/2})}\frac{\boldsymbol{u}_g - \boldsymbol{u}_d^{(j)}}{\theta\overline{S}}. \end{aligned} \tag{3.59}$$

where $\overline{S} = (S_{j+1} + S_j)/2$ is the mean surface area of the droplets in the section j, used to evaluate the averaged drag force in this section. The evaporation terms $E_1^{(j)}$ and $E_2^{(j)}$ write:

$$E_1^{(j)} = \frac{5S_j^{3/2}}{2(S_{j+1}^{5/2} - S_j^{5/2})}\mathrm{R}_S S_j \tag{3.60}$$

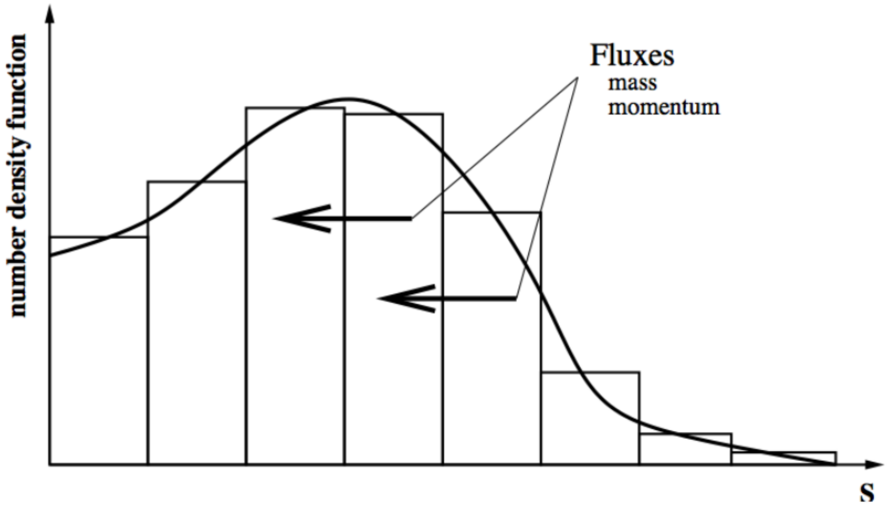


Figure 3.3: *Size distribution with the Multi-Fluid method: the arrows show the evaporation and momentum fluxes from a section to another.*

and

$$E_2^{(j)} = \frac{5S_j^{3/2}}{2(S_{j+1}^{5/2} - S_j^{5/2})} \int_{S_j}^{S_{j+1}} \frac{3}{2}\sqrt{S}\mathrm{R}_S(S)dS. \tag{3.61}$$

The last term can be simplified further in the case of a constant evaporation rate. The system of equation (3.58) can be seen as a finite volume discretization in size variable, as illustrated in figure 3.3.

The OSM is a first order accurate model in the size discretization ΔS, which leads to a high numerical diffusion and requires a great number of sections to counterbalance its drawbacks. The Two-Size moments (TSM) method, where two size-moments are used by section, was proposed in Laurent(2006) to increase the accuracy. In this case, linear (Aff-TSM) Laurent(2006); Laurent et al.(2016) or exponential reconstruction (Exp-TSM) Dufour(2005) of the size distribution are envisioned to close the model. The TSM method has been tested and shown a good accuracy in the evaluation of the evaporation and the drag force, since it provides more accurate information at the section boundaries. Finally, we underline that a combined TSM (for the size distribution) and Anisotropic Gaussian (for the velocity distribution) model has been recently developed to take into account both the polydispersion and the polykinetic features of realistic spray configurations Boileau et al.(2016).

3.4.2.2 High order size moments

A second idea to capture the polydispersion, without need to discretize the size space into several sections, consists in transporting high order size moments. This method was first introduced in Massot et al.(2010); Kah et al.(2012), where the authors have developed the EMSM model under a monokinetic assumption:

$$f(t, \boldsymbol{x}, \boldsymbol{c}, S) = n(t, \boldsymbol{x}, S)\delta(\boldsymbol{c} - \boldsymbol{u}_d(t, \boldsymbol{x})). \tag{3.62}$$

In this section, we recall briefly the main concept of this approach, while more details are given in the next chapter. Considering the presumed NDF (3.62), we define its k-th size moments as follows:

$$m_k = \int_0^1 S^k n(S) dS. \tag{3.63}$$

Then a system of equations for a set of N moments and 1 momentum to solve for the velocity field, can be derived from the WB equation (3.41):

$$\begin{array}{rcl}
\partial_t m_0 + \partial_{\boldsymbol{x}}.(m_0 \boldsymbol{u}_d) & = & \psi_+^0 - \psi_-^0, \\
\partial_t m_1 + \partial_{\boldsymbol{x}}.(m_1 \boldsymbol{u}_d) & = & \psi_+^1 - \psi_-^1 - \mathrm{K}\ m_0, \\
\partial_t m_2 + \partial_{\boldsymbol{x}}.(m_2 \boldsymbol{u}_d) & = & \psi_+^2 - \psi_-^2 - 2\mathrm{K}\ m_1, \\
\vdots & & \vdots \\
\partial_t m_{N-1} + \partial_{\boldsymbol{x}}.(m_{N-1} \boldsymbol{u}_d) & = & \psi_+^{N-1} - \psi_-^{N-1} - (N-1)\mathrm{K}\ m_{N-2}, \\
\partial_t (m_1 \boldsymbol{u}) + \partial_{\boldsymbol{x}}.(m_1 \boldsymbol{u}_d \otimes \boldsymbol{u}_d) & = & -\mathrm{K}\ m_0 \boldsymbol{u}_d + (\psi_+^1 - \psi_-^1)\boldsymbol{u}_d \\
 & & + m_0 \dfrac{\boldsymbol{u}_g - \boldsymbol{u}_d}{\theta},
\end{array} \tag{3.64}$$

where $\psi_-^k = S_{min}^k n(S_{min})$ (resp. $\psi_+^k = S_{max}^k n(S_{max})$) is the instantaneous disappearance flux (resp. appearance flux) coming from other sections. Here, we choose to write the system of equations using the size-section $[S_{min}, S_{max}]$ instead of $[0,1]$, in order the possibility to extend the model to an hybrid high order size-moment and sectional model. This hybrid model satisfies similar equations as in (3.64) for each size section by substituting the interval $[S_{min}, S_{max}]$ by $[S_j, S_{j+1}]$ for $1 \leq j \leq N_s$.

In the case of single section, we have $[S_{min}, S_{max}] = [0,1]$. In this case, we have $n(S_{max} = 1) = 0$ then for all $k \geq 0$ $\psi_+^k = 0$. And since $S_{min} = 0$ then for all $k > 0$ $\psi_-^k = 0$ and $\psi_-^0 = n(S = 0)$. The Stokes number has been supposed to depend linearly on the droplet surface: $\mathrm{St}(S) = \theta S$, where $\theta = \tau_p(S_0)/\tau_g$ and S_0 is the reference size variable used in 3.3.5, τ_p is defined in (3.23). We choose to write the system of equation using the size-section $[S_{min}, S_{max}]$ instead of $[0,1]$, to show that we can also extend the model in a combined EMSM and multi-fluid approach, where we replace $[S_{min}, S_{max}]$ by $[S_j, S_{j+1}]$ for each size-section $0 \leq j \leq N_s$ (see the definition of size-section in the previous section).

Yet, the system is not fully closed because the value $n(S = 0)$ is unknown. It can be interpreted as the instantaneous droplets disappearance flux by evaporation.

To close the system, we need a smooth reconstruction of the NDF $n(S)$ and this is done by the maximization of the following Shannon entropy:

$$H(n) = -\int_0^1 n(s) \ln(n(s)) ds. \tag{3.65}$$

The existence and uniqueness of a density function n which maximizes the Shannon entropy and satisfies:

$$\begin{array}{rcl}
m_0 & = & \int_0^1 n(S) dS, \\
\vdots & & \vdots \\
m_{N-1} & = & \int_0^1 S^{N-1} n(S) dS,
\end{array} \tag{3.66}$$

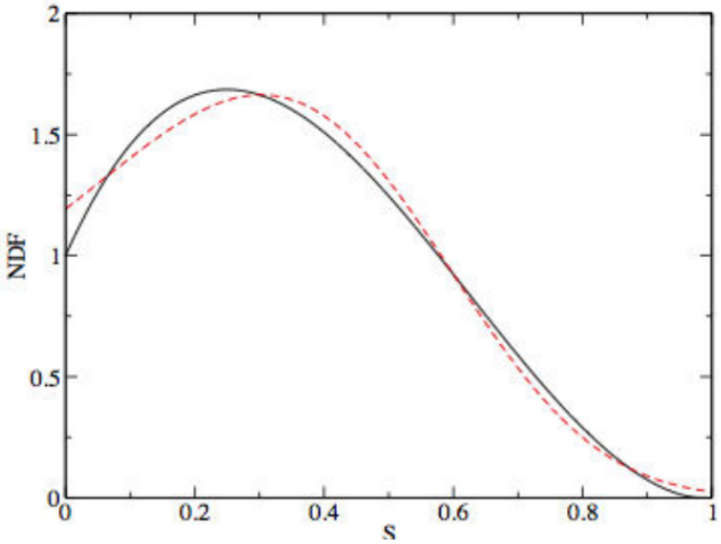


Figure 3.4: *Reconstruction of the size distribution through entropy maximisation (red dashed line), the real size distribution (black solid line).*

was proved in Mead and Papanicolaou(1984), and the solution is shown to have the following form:

$$n(S) = exp\left(-(\lambda_0 + \lambda_1 S + \ldots + \lambda_{N-1} S^{N-1})\right), \tag{3.67}$$

where coefficients λ_k, are determined from the system (3.66). In the same article, the authors propose an algorithm to solve this constrained optimization problem, based on a Newton-Raphson method. Figure 3.4 illustrates a reconstructed size-distribution through entropy maximisation and using the first four order moments of a real size distribution.

The resolution of the transport equation of high order moments with source terms needs special treatment due to the complexity of the moment space. These issues will be discussed in more details in chapters 3, 6 and 7.

3.5 Conclusion

A general framework of an Eulerian spray modeling based on a kinetic approach is presented in this chapter. The gaseous phase (carrier phase) is described by the Navier Stokes equations including species transport equations to take into account the reactive character of the flow. In the case of a dilute spray, the interaction with the disperse phase can be taken into account through source terms. A description of the disperse phase at a mesoscopic level is given by the Williams-Boltzmann equation. From this equation, an hierarchy of models can be derived based on the moments method. The size-velocity moments order are discussed along this chapter. The choice of the model and of the closure hypotheses depend on the level of the description needed to capture the polydispersion or/and the polykinetic features of the spray. For the polydispersion, two types of models are proposed: the multi-fluid models and high order size-moment models (EMSM). The EMSM model shows to be a promising method to capture the polydisperion of the spray without the need of a fine size discretization. Furthermore, a hybrid method that combines EMSM and multi-fluid would allow to capture the size-velocity correlation using a small number of size-sections. Another way to capture the size-velocity was proposed by Vié et al Vié et al.(2013b). The authors developed the CSVM to extend the EMSM model and to take into account the size-velocity correlation using only one size-section. In the following, we adopt the high order size-moments approach to model the spray and capture the polydispersion. However, the size-moments used in the EMSM model are restricted to describe the polydispersion of spherical droplets. We need to keep in mind that the disperse phase is obtained from the atomization of the fuel in the separated phases zone, where the gas-liquid interface geometry has an important influence on the atomization process. It is then important to find a set of the size-moments that can be extended and related to some intrinsic variables in the separated phases.

Part II

Contribution to a unified modeling of disperse and separated phases

Chapter 4

High order fractional moment for evaporating sprays: toward a geometrical description

4.1 Introduction

As it was discussed in the general introduction, one of the main difficulties in fuel injection modeling consists in designing a model with a unified description for the separated phases and the disperse phase zones. In chapters 2 and 3, we have presented two families of reduced order models, which can be used separately in each of the two zones. However, none of these two models is suitable to the simulation of the whole injection process. Separated two-fluid models (chapter 2) suffer from a high artificial mixing near the interface, which as a consequence, leads to interface geometry loss. On the other hand, Eulerian spray models (see chapter 3) suffer from the inefficiency in describing complex polydispersion. Even if the EMSM and the multi-fluid models have shown promising results in modeling the polydispersion of spherical droplets and evaluating accurately the evaporation fluxes, an initial size distribution is required to launch the simulations. In the present work, we propose to resolve the polydispersion by using a set of size-moments, which can be identified as averages of the gas-liquid interface geometry variables, Drew(1990). Indeed, we show that some geometrical variables can be expressed as fractional size-moments of the NDF.

In this chapter, we introduce the mathematics fundamentals of the new fractional moments model. We show that we can preserve the advantages of the previous methods Kah et al.(2012); Massot et al.(2010) in terms of both mathematical properties and numerical tools, but with a much higher potential in terms of coupling with a sub-scale description of a complex interface. First, we recall some properties of the moments space in the case of integer moments, before generalizing these results to the case of fractional moments. The evaporation flux is evaluated using Maximum Entropy (ME) reconstruction based on fractional moments[1]. The existence and uniqueness of this convex optimization problem under constraints are given in Mead and Papanicolaou(1984) in the case of integer moments. While some elements of the proof for the

---

[1] Entropy maximization with fractional moments is used in Inverardi et al.(2003); Gzyl and Tagliani(2010) as a remedy to ill-conditioning related to a high number of integer moments Talenti(1987). The considered set of fractional moments is then recovered from the integer ones, and their orders are optimized to minimize the entropy difference with the real function. In the present contribution, a known and small set of fractional moments is used, deduced from physical considerations, in such a way that the problematic is different.

fractional moments case can be found in Kapur and Kesavan(1992), we propose a complete proof in the case of a special set of fractional moments. Moreover, we generalize useful properties to the fractional moment space, such as canonical moments, as well as lower principal representation, Dette and Studden(1997). These properties are relevant ingredients to design high order realizable schemes and algorithms to solve a high order moments system. These results will be directly used in chapters 7 and 8 to design robust and accurate numerical schemes for the model proposed in this chapter.

4.2 Integer high order size moments modeling and mathematical properties

In this part, we discuss the mathematical properties of the EMSM model presented briefly in 3.4.2.2. In this model, a set of integer size-moments is used to capture the polydispersion, where the size is given by the surface area of the droplets:

$$\mathrm{m}_k = \int_0^1 S^k n(S) dS. \tag{4.1}$$

We recall that here, as in the previous chapter, we use dimensionless variables ($S_{max} = 1$). The set of integer moments satisfies the following system of equation:

$$\begin{array}{rcll} \partial_t \mathrm{m}_0 + \partial_{\boldsymbol{x}}.(\mathrm{m}_0 \boldsymbol{u}) & = & -\mathrm{K}n(t,\boldsymbol{x},S=0) & , \\ \partial_t \mathrm{m}_1 + \partial_{\boldsymbol{x}}.(\mathrm{m}_1 \boldsymbol{u}) & = & -\mathrm{K}\ \mathrm{m}_0 & , \\ \vdots & & \vdots & \\ \partial_t \mathrm{m}_N + \partial_{\boldsymbol{x}}.(\mathrm{m}_N \boldsymbol{u}) & = & -N\mathrm{K}\ \mathrm{m}_{N-1} & , \\ \partial_t (\mathrm{m}_1 \boldsymbol{u}) + \partial_{\boldsymbol{x}}.(\mathrm{m}_1 \boldsymbol{u} \otimes \boldsymbol{u}) & = & -\mathrm{K}\ \mathrm{m}_0 \boldsymbol{u} & +\mathrm{m}_0 \dfrac{\boldsymbol{u}_g - \boldsymbol{u}}{\theta}. \end{array} \tag{4.2}$$

Here again, we use the same notations as in the previous chapter. The unclosed term $-\mathrm{K}n(t,\boldsymbol{x},S=0)$ expresses the pointwise disappearance flux of droplets through evaporation. Even though, this term is involved only in the first equation, it contributes to the evolution of the other moments in the same way through the terms $-k\mathrm{K}\mathrm{m}_{k-1}$ as shown in Massot et al.(2010). In the following, we present the definition of the moment space and some useful properties in the case of integer moments. Then, we discuss the continuous reconstruction of the size distribution used in the work of Kah(2010) to close the EMSM model.

4.2.1 Integer moment space and properties

We denote by $P([0,1])$ the set of all probability density measures of the interval $[0,1]$. Then, the Nth "normalized" moment space $\mathbb{M}_N$ is defined as follows:

$$\mathbb{M}_N = \{\boldsymbol{c}_N(\mu), \mu \in P\}, \ \ \boldsymbol{c}_N = (c_0(\mu), \dots, c_N(\mu))^t, \ \ c_k(\mu) = \int_0^1 x^k \mu(x) dx. \tag{4.3}$$

Let us notice that $c_0 = 1$, since we use probability density measures. In our case, we can normalize by m_0 to associate each moment vector $(\mathrm{m}_0, \dots, \mathrm{m}_N)^t$ to a normalized moment vector $(c_0, \dots, c_N)^t \in \mathbb{M}_N$, where $c_k = \mathrm{m}_k/\mathrm{m}_0$.

Definition The N^{th} moment space is defined as the set of vectors $\mathbf{m}$, for which the normalized vector by m_0 belongs to the N^{th} normalized moment space, $\mathbb{M}_N$.

The characterization of the moment space can be done through the characterization of the normalized moment space. The normalized moment space is a convex and bounded space but it has a complex geometry. The major challenges for numerical simulations of a moment model consist in designing numerical schemes, which can ensure that the moment vector $\boldsymbol{c}_N$ belongs to $\mathbb{M}_N$ at all time. For this reason, it is important to characterize this space. The positivity of Hankel-Hadamard determinants is a necessary and sufficient condition for the vector $\boldsymbol{c}_N$ to be in the normalized moment space $\mathbb{M}_N$. As it was proposed in Dette and Studden(1997), these conditions read as follows:

Proposition 4.2.1 *$\boldsymbol{c}_N$ belongs to $\mathbb{M}_N$ if and only if for all $d = 0, 1$ and $m \geq 0$, we have* $\det(\underline{H}_{2m+d}) \geq 0$ *and* $\det(\overline{H}_{2m+d}) \geq 0$*, where:*

$$\underline{H}_{2m+d} = \begin{pmatrix} c_d & \dots & c_{m+d} \\ \vdots & & \vdots \\ c_{m+d} & \dots & c_{2m+d} \end{pmatrix} \quad \overline{H}_{2m+d} = \begin{pmatrix} c_{1-d} - c_{2-d} & \dots & c_m - c_{m+1} \\ \vdots & & \vdots \\ c_m - c_{m+1} & \dots & c_{2m-1+d} - c_{2m+d} \end{pmatrix} \tag{4.4}$$

Canonical moments: the moment space can also be characterized by the canonical moments Dette and Studden(1997). The geometry of the space of the canonical moments vectors is much simpler since it is the cube $[0, 1]^N$. Let us consider a normalized moment vector $\boldsymbol{c}_N = (c_0, \dots, c_N)^t$. We denote by $P(\boldsymbol{c}_N) \subset P([0, 1])$ the set of all probability density measures, whose first $N + 1$ moments are given by the vector $\boldsymbol{c}_N$:

$$\boldsymbol{c}_N = \int_0^1 \begin{pmatrix} 1 \\ S \\ \vdots \\ S^N \end{pmatrix} \mu(S) dS. \tag{4.5}$$

As long as $\boldsymbol{c}_{N-1}$ belongs to the interior of $\mathbb{M}_{N-1}$, the canonical moment vector $\boldsymbol{p}_N = (p_1, \dots, p_N)^t$ associated to $\boldsymbol{c}_N$ can be defined as follows:

$$p_k = \frac{c_k - c_k^-(\boldsymbol{c}_{k-1})}{c_k^+(\boldsymbol{c}_{k-1}) - c_k^-(\boldsymbol{c}_{k-1})}, \tag{4.6}$$

where $1 \leq k \leq N$ and the variables $c_k^-(\boldsymbol{c}_{k-1})$ and $c_k^+(\boldsymbol{c}_{k-1})$ depend on the moment vector of order $k - 1$, $\boldsymbol{c}_{k-1} \in \mathbb{M}_{k-1}$ as follows:

$$c_k^-(\boldsymbol{c}_{k-1}) = \min_{\mu \in P(\boldsymbol{c}_{k-1})} c_k(\mu), \quad c_k^+(\boldsymbol{c}_{k-1}) = \max_{\mu \in P(\boldsymbol{c}_{k-1})} c_k(\mu). \tag{4.7}$$

Then, a vector $\mathbf{m}_N$ belongs to the moment space if and only if the associated canonical moment vector belongs to the compact space $[0, 1]^N$. Also it is in the interior of the moment space if and only if the canonical moment vector is in $]0, 1[^N$.

An algorithm, given in Dette and Studden(1997), allows to go from moments to canonical moments and conversely. In the case of moments up to order three, we can use the following algebraic expressions:

$$\begin{aligned} p_1 &= \frac{\mathrm{m}_1}{\mathrm{m}_0}, \\ p_2 &= \frac{\mathrm{m}_0\mathrm{m}_2 - \mathrm{m}_1^2}{(\mathrm{m}_1 - \mathrm{m}_0)\mathrm{m}_1}, \\ p_3 &= \frac{(\mathrm{m}_1 - \mathrm{m}_0)(\mathrm{m}_1\mathrm{m}_3 - \mathrm{m}_2^2)}{(\mathrm{m}_1 - \mathrm{m}_0)(\mathrm{m}_0\mathrm{m}_2 - \mathrm{m}_1^2)}. \end{aligned} \tag{4.8}$$

Lower and upper principal representation The Hausdorff finite moment problem associated to $\boldsymbol{c}_N$ consists in finding a positive distribution $\mu \in P([0,1])$ such that:

$$c_k = \int_0^1 x^k \mu(x)\mathrm{d}x. \tag{4.9}$$

For a moment vector $\boldsymbol{c}_N$ lying in the interior of the normalized moment space, it has an infinity of solutions. In this condition, we also have $c_{N+1}^- < c_{N+1}^+$ and the two moments vectors $\boldsymbol{c}_{N+1}^-(\boldsymbol{c}_N) = (c_0, \dots, c_N, c_{N+1}^-)^t$ and $\boldsymbol{c}_{N+1}^+(\boldsymbol{c}_N) = (c_0, \dots, c_N, c_{N+1}^+)^t$ belong to the boundary of the normalized moment space. Now, the Hausdorff finite moment problem associated to these moments of order $N+1$ has a unique solution. We call lower (respectively upper) principal representation, the distribution μ^- (respectively μ^+) which is the solution of the Hausdorff problem associated to $\boldsymbol{c}_{N+1}^-(\boldsymbol{c}_N)$ (respectively $\boldsymbol{c}_{N+1}^+(\boldsymbol{c}_N)$). The two distributions μ^- and μ^+ can be written as a sum of weighted delta Dirac distributions and are also two different solutions of the Hausdorff problem associated to $\boldsymbol{c}_N$. The Product-Difference (PD) algorithm Gordon(1968) allows to compute the abscissas and weight of the lower principal representations. This algorithm consists in finding weights $(w_i)_{i\in[1,s]}$ and abscissas $(S_i)_{i\in[1,s]}$, such that:

$$\mathbf{m} = \begin{pmatrix} \mathrm{m}_0 \\ \mathrm{m}_1 \\ \vdots \\ \mathrm{m}_N \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^s w_i \\ \sum_{i=1}^s w_i S_i \\ \vdots \\ \sum_{i=1}^s w_i S_i^N \end{pmatrix}, \tag{4.10}$$

where $N = 2s - 1$. This algorithm is used to evaluate the decreasing droplet sizes due to evaporation for the EMSM model Massot et al.(2010).

4.2.2 Continuous reconstruction of the size distribution

Massot et al.(2010) proposed to use a continuous reconstruction of the size distribution through the maximization of the Shannon entropy:

$$H(n) = -\int_0^1 n(S)\ln(n(S))dS. \tag{4.11}$$

The existence and uniqueness of a size distribution $n^{ME}(S)$ that maximizes the Shannon entropy and is the solution of the finite Hausdorff moment problem (3.66) was proved in Mead and Papanicolaou(1984) for the moments of integer order, when the moment vector belongs to the interior of the moment space. The solution is shown to have the following form:

$$n(S) = \exp\left(-(\lambda_0 + \lambda_1 S + \dots + \lambda_N S^N)\right), \tag{4.12}$$

where the coefficients λ_i are determined by system (3.66). The resolution of this nonlinear problem can be achieved by using Newton-Raphson method. The limitation of this algorithm at the moment space boundary, or equivalently when the ME reconstructed size-distribution approaches a sum of Dirac distributions, is discussed in Massot et al.(2010). Vié et al.(2013b) proposed some more advanced solutions to cope with this problem, by tabulating the coefficients depending on the moments and by using an adaptive support for the integral calculation. These techniques have an accurate computation of the integral moments, when the NDF is nearly singular and accelerate the convergence of the Newton-Raphson algorithm.

4.2.3 Limitation of the integer moments to a disperse phase model

Even though this high order moment formalism provides some key information about the polydispersion using only one size-section, it is important to realize that its use is restricted to the disperse phase zone. Coupling such an approach with a separated phases model requires some complementary information, which two-fluid models can usually not provide. Indeed, separated two-fluid models consider the interface as a smooth transition layer, within which important information about the interface geometry have been lost. A first step would be to enrich the separated two-fluid model as in Drui et al.(2016b) in order to transport averaged geometrical variables to gain accuracy about the interface geometry. Nevertheless, even in the framework of such an enriched two-fluid model, the coupling of two very different models is usually a rather cumbersome task and relies on parameters the described physics will depend on.

Consequently, we adopt an original strategy and build a high order moment model for the disperse phase. This new approach possesses the same key properties as the EMSM model in terms of accuracy, robustness and computational cost. But, it involves a different set of variables that describes the averaged interface geometry, so that we end up with a set of common variables in the two zones. This can potentially help in building a single unified model able to capture the proper physics in both zones. In order to introduce the new set of variables, we first have to recall the natural geometrical variables in the separated phases zone, before extending this description to the disperse phase.

4.3 Improvement of the gas-liquid interface description in two-fluid models

In chapter 2, we explain that in the separated phases, one can use two-fluid models when the interface does not need to be resolved. However, this type of model spreads the interface. This diffusion can erase some important two-phase flow structures and thus, leads to an inaccurate prediction of the atomization in the intermediate zone of the fuel injection. Some contributions in the literature proposed to limit the diffusion of the interface, either by using numerical solutions or by adding some anti-diffusion source terms (see chapter 2 for more details). In the present work, we focus in modeling the interface sub-scales, when the interface is strongly deformed and no more captured by two-fluid models. We interpret the zone where the interface is diffused as an ensemble of interface realizations. In this representation, the interface can be described statistically using some averaged variables. In chapter 2, we have presented some works on the modeling of primary atomization, where the gas-liquid interface description is enriched with an additional surface area density variable. In this chapter, we step further in enriching the gas-liquid interface using the geometrical averaged variables proposed by Drew(1990).

4.3.1 Definition of the geometrical variables of the gas-liquid interface

In the following, we use the same notations as the one introduced in chapter 2. We recall that the DIMs are obtained by averaging the monophasic fluid equations. The obtained equations involve the volume fraction, interpreted as the volume portion occupied by a given phase. This variable allows to locate the interface up to the averaging scale and is then a first piece of information about the interface geometry:

$$\alpha_k(t,\boldsymbol{x}) = \langle \chi_k(t,\boldsymbol{x}') \rangle \,, \tag{4.13}$$

where χ_k is the presence function (see equation (2.4)).

The second variable considered in Drew(1990) and used also in other two-phase flow models Vallet et al.(2001); Lebas et al.(2009) is the interfacial area density. This variable is important in the modeling of exchange terms (evaporation, thermal transfer and drag force) as well as primary atomization. It is interpreted as the ratio of the surface area of an interface to the volume it occupies.

$$\Sigma(t,\boldsymbol{x}) = \langle ||\boldsymbol{\nabla}\chi_k(t,\boldsymbol{x}')|| \rangle \,. \tag{4.14}$$

So far, the interface description is still incomplete, since no information on the interface shape has been given. In fact, the small details of the interface can not be modeled accurately using only two geometrical variables. Drew proposed to introduce the mean curvature $H = \frac{1}{2}(\mathrm{k}_1 + \mathrm{k}_2)$ and the Gauss curvature $G = \mathrm{k}_1\mathrm{k}_2$ of the interface, where k_1 and k_2 are the two principal curvatures (a rigorous definition of these curvatures can be found in chapter 5). These variables are defined only at the interface. Therefore, we need a specific averaging for these interfacial variables. To do so, we introduce the interfacial averaging operator $\widetilde{(\bullet)}$, defined as follows:

$$\Sigma\,\widetilde{(\bullet)}(t,\boldsymbol{x}) = \langle (\bullet)||\boldsymbol{\nabla}\chi_k(t,\boldsymbol{x}')|| \rangle \tag{4.15}$$

As an example, the interfacial averaged Gauss and mean curvatures are defined as follows :

$$\begin{aligned} \Sigma\widetilde{H}(t,\boldsymbol{x}) &= \langle H||\boldsymbol{\nabla}\chi_k(t,\boldsymbol{x}')|| \rangle \,, \\ \Sigma\widetilde{G}(t,\boldsymbol{x}) &= \langle G||\boldsymbol{\nabla}\chi_k(t,\boldsymbol{x}')|| \rangle \,. \end{aligned} \tag{4.16}$$

4.3.2 Transport equations of the geometrical variables

From a kinematic evolution of the surface, Drew(1990) derived evolution equations of the two curvatures. In this derivation, Drew considered a normal interface velocity $\mathbf{v}_I = \mathrm{v}_I\boldsymbol{n}$, where $\boldsymbol{n}$ is the normal vector at the surface, oriented toward the gas phase. These equations read as follows:

$$\begin{aligned} \frac{d^I H}{dt} &= -\frac{1}{2}\nabla_I^2(\mathrm{v}_I) - (2H^2 - G)\mathrm{v}_I, \\ \frac{d^I G}{dt} &= -H\nabla_I^2(\mathrm{v}_I) + \sqrt{H^2 - G}\Big(\frac{\partial^2 \mathrm{v}_\mathrm{I}}{\partial {y_1^*}^2} - \frac{\partial^2 \mathrm{v}_I}{\partial {y_2^*}^2}\Big) - 2HG\mathrm{v}_I, \end{aligned} \tag{4.17}$$

where $\dfrac{d^I \bullet}{dt} = \partial_t \bullet + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\bullet)$ is a Lagrangian time derivative following an interface point.The first terms appearing on the right-hand side of equations (5.15) involve a surface Laplace operator. The superscript $*$ refers to the principal direction of curvature (see chapter 5 for more details).

By using the evolution equation on the presence function (2.5), we can derive an equation on the Instantaneous Area concentration (IAC) variable $||\nabla_{\boldsymbol{x}}(\chi_k)||$:

$$\frac{d^I||\nabla_{\boldsymbol{x}}(\chi_k)||}{dt} = (\mathrm{v}_I H - \nabla_I \cdot (\mathbf{v}_I))\, ||\nabla_{\boldsymbol{x}}(\chi_k)||. \tag{4.18}$$

By combining the evolution equation on the curvatures (4.17) and the evolution equation on (4.18), we obtain the following equations:

$$\begin{aligned}
\partial_t||\nabla_{\boldsymbol{x}}(\chi_k)|| + \nabla_{\boldsymbol{x}} \cdot (||\nabla_{\boldsymbol{x}}(\chi_k)||\mathbf{v}_I) &= \mathrm{v}_I H||\nabla_{\boldsymbol{x}}(\chi_k)|| \\
\partial_t H||\nabla_{\boldsymbol{x}}(\chi_k)|| + \nabla_{\boldsymbol{x}} \cdot (H||\nabla_{\boldsymbol{x}}(\chi_k)||\mathbf{v}_I) &= -\frac{1}{2}\nabla_I^2(\mathrm{v}_I)||\nabla_{\boldsymbol{x}}(\chi_k)|| - (H^2 - G)\mathrm{v}_I||\nabla_{\boldsymbol{x}}(\chi_k)|| \\
\partial_t G||\nabla_{\boldsymbol{x}}(\chi_k)|| + \nabla_{\boldsymbol{x}} \cdot (G||\nabla_{\boldsymbol{x}}(\chi_k)||\mathbf{v}_I) &= -H\nabla_I^2(\mathrm{v}_I)||\nabla_{\boldsymbol{x}}(\chi_k)|| + \\
&\quad \sqrt{H^2 - G}\left(\frac{\partial^2 \mathrm{v}_I}{\partial {y_1^*}^2} - \frac{\partial^2 \mathrm{v}_I}{\partial {y_2^*}^2}\right)||\nabla_{\boldsymbol{x}}(\chi_k)|| - HG\mathrm{v}_I||\nabla_{\boldsymbol{x}}(\chi_k)||.
\end{aligned} \tag{4.19}$$

In Drew(1990), transport equations of the averaged geometrical variables $(\alpha_k, \Sigma, \Sigma\widetilde{H}, \Sigma\widetilde{H})$ have been derived by averaging these equations and the kinematic equation of the presence function (2.5) using the ensemble averaging operator. However, the final set of equations involve source terms that are unclosed. In chapter 5, we show how to derive similar equations using a different approach that can provide us by more details and understanding of the unclosed source terms. In the following of this chapter, we simplify the modeling framework by considering spherical droplets, then we derive the equations of these averaged geometrical quantities.

4.4 Geometrical high order moments model

4.4.1 Interfacial geometrical variables for the disperse phase

Let us consider a population of spherical droplets represented by their size distribution $n(t, \boldsymbol{x}, S)$. Then, by analogy with the separated phases, we wish to express the averaged geometrical variables: the volume fraction, the interface area density, the Gauss curvature and the mean curvature in the disperse phase. In the separated case, the definition of these geometrical variables is based on the phase function χ_k. This function contains all the microscopic information about the interface. In the disperse phase, we use the statistical information about the droplets distribution, which is given by the size distribution $n(t, \boldsymbol{x}, S)$. Considering this function, we define these different geometrical variables in the context of a polydisperse spray as follows:

A. The *volume fraction* α_d is the sum of the volume of each droplet divided by the occupied volume at a given position. Since $n(t, \boldsymbol{x}, S)$ is a number of droplets per unit volume and surface area, the volume fraction writes:

$$\alpha_d = \int_0^1 V(S)n(t, \boldsymbol{x}, S)dS. \tag{4.20}$$

The droplet being spherical, $V(S) = \dfrac{S^{3/2}}{6\sqrt{\pi}}$.

B. The *interfacial area density* Σ_d is the sum of the surface of each droplet divided by the occupied volume at a given position:

$$\Sigma_d = \int_0^1 Sn(t, \boldsymbol{x}, S)dS. \tag{4.21}$$

C. For a spherical droplet, the two principal curvatures are equal everywhere on the droplet surface: $\mathrm{k}_1 = \mathrm{k}_2 = \frac{2\sqrt{\pi}}{\sqrt{S}}$. But since we use the mean and Gauss curvatures, we can define two different averaged quantities. Let us notice that, in the case of separated phases, the *average mean and Gauss curvatures* were defined as an average over a volume and weighed by the interfacial area. In the disperse phase case, this becomes:

$$\begin{aligned} \Sigma_d \widetilde{H}_d &= \int_0^1 H(S) S n(t,\boldsymbol{x},S) dS, \\ \Sigma_d \widetilde{G}_d &= \int_0^1 G(S) S n(t,\boldsymbol{x},S) dS, \end{aligned} \tag{4.22}$$

where $H(S) = \frac{2\sqrt{\pi}}{\sqrt{S}}$ and $G(S) = \frac{4\pi}{Sd}$. These four geometrical variables can be expressed as fractional moments of the size distribution:

$$\begin{aligned} \Sigma_d \widetilde{G}_d &= 4\pi \mathrm{m}_{0/2}, \\ \Sigma_d \widetilde{H}_d &= 2\sqrt{\pi}\mathrm{m}_{1/2}, \\ \Sigma_d &= \mathrm{m}_{1/2}, \\ \alpha_d &= \frac{1}{6\sqrt{\pi}}\mathrm{m}_{3/2}, \end{aligned} \tag{4.23}$$

where $\mathrm{m}_{k/2} = \int_0^1 S^{k/2} n(S) dS$.

In fact, these moments can be expressed as integer moments by a simple variable substitution: $x = \sqrt{S}$. However, we prefer to hold the droplet surface as the size variable, since we consider a d^2 evaporation law, for which the evaporation rate $\mathrm{R}_S = \dfrac{dS}{dt}$ is constant.

4.4.2 The governing moment equation

In this section, we derive from the kinetic equation (3.41), in the same way as it was done in the derivation of the EMSM model, a high order fractional moment model. This model gives the evolution of the average geometrical interfacial variables due to transport, evaporation and drag force. The system reads:

$$\begin{cases} \partial_t \mathrm{m}_{0/2} + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{0/0}\boldsymbol{u}) = -\mathrm{K}n(t,\boldsymbol{x},S=0), \\ \partial_t \mathrm{m}_{1/2} + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{1/2}\boldsymbol{u}) = -\dfrac{\mathrm{K}}{2}\mathrm{m}_{-1/2}, \\ \partial_t \mathrm{m}_{2/2} + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{2/2}\boldsymbol{u}) = -\mathrm{K}\mathrm{m}_{0/2}, \\ \partial_t \mathrm{m}_{3/2} + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{3/2}\boldsymbol{u}) = -\dfrac{3\mathrm{K}}{2}\mathrm{m}_{1/2}, \\ \partial_t (\mathrm{m}_{2/2}\boldsymbol{u}) + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{2/2}\boldsymbol{u} \otimes \boldsymbol{u}) = -\mathrm{K}\mathrm{m}_{0/2}\boldsymbol{u} + \mathrm{m}_{0/2}\dfrac{\boldsymbol{u}_g - \boldsymbol{u}}{\theta}, \end{cases} \tag{4.24}$$

where $-\mathrm{K}n(t,\boldsymbol{x},S=0)$ represents the pointwise disappearance flux, and the moment of negative order, $\mathrm{m}_{-1/2} = \int_0^1 S^{-1/2} n(t,S) dS$, naturally appears in the system after integrating by part the evaporation term in the WBE. In the following, these terms and the associated instantaneous fluxes will be closed by a smooth reconstruction of the size distribution through entropy maximization.

The use of fractional moments introduces a new mathematical framework of modeling as well as some numerical difficulties, which require specific treatments. In the following, we generalize the mathematical properties, presented in 4.2.1 for integer moments, to the case of fractional moments. These results will be used to design realizable numerical schemes in chapters 7 and 8, i.e. schemes that ensure the preservation of the moment vector in the moment space.

4.4.3 Fractional moments space

The purpose of this part is to extend the definition and the properties of the integer moments space to the fractional moments space, in order to provide necessary tools to characterize the topology of this space. This will be of paramount importance to design realizable numerical schemes. In the following, we use the normalized fractional moments $\mathrm{c}_{k/2} = \mathrm{m}_{k/2}/\mathrm{m}_{0/2}$.

Definition We define the Nth normalized fractional moment space as follows:

$$\mathbb{M}_N^{1/2}([0,1]) = \{\mathbf{c}_{N/2}(\mu), \quad \forall \mu \in \mathrm{P}\left([0,1]\right)\}, \; \mathbf{c}_{N/2}(\mu) = (\mathrm{c}_{0/2}(\mu), \mathrm{c}_{1/2}(\mu), \dots, \mathrm{c}_{N/2}(\mu))^t,$$

where $\mathrm{P}([0,1])$ denotes the set of probability density measures defined on the interval $[0,1]$ and $\mathrm{c}_{k/2} = \int_0^1 x^{k/2}\mu(x)dx$.

The fractional moments can be expressed as integer moments by using the following substitution $r^2 = x$:

$$\mathrm{c}_{k/2}(\mu) = \int_0^1 x^{k/2}\mu(x)dx = \int_0^1 r^k \underbrace{(2r\mu(r^2))}_{\widetilde{\mu}(r)} dr. \tag{4.25}$$

This relation expresses an identification between the fractional moment $\mathrm{c}_{k/2}(\mu)$ of the measure μ and the integer moment $\widetilde{\mathrm{c}}_k = \mathrm{c}_k(\widetilde{\mu})$ of the measure $\widetilde{\mu}(r) = 2r\mu(r^2)$. In the following, we use these notations to differentiate between the two types of the moments. With this simple identification, we will take benefit from the already existing results on the integer moment space, and extend them to the case of fractional moments.

Canonical fractional moments for a fractional moment vector $\mathbf{c}_{N/2} \in \mathbb{M}_N^{1/2}([0,1])$, we denote by $\mathrm{P}_N(\mathbf{c}_{N/2})$ the set of all measures $\mu \in \mathrm{P}([0,1])$, which are the solution of the following finite Hausdorff problem:

$$\mathrm{c}_{k/2} = \int_0^1 x^{k/2}\mu(x)dx, \quad k/2 \leq N/2. \tag{4.26}$$

If $\mathbf{c}_{N/2} = (\mathrm{c}_{0/2}, \dots, \mathrm{c}_{N/2})^t$ belongs to the interior of $\mathbb{M}_N^{1/2}([0,1])$, we can show that the set $\mathrm{P}_N(\mathbf{c}_{N/2})$ is infinite. Indeed, these results were shown in Dette and Studden(1997) for integer moments, and its generalization to fractional moments is straightforward thanks to the identification (4.25). Furthermore, the set of $\mathrm{c}_{(N+1)/2}(\mu)$, where $\mu \in \mathrm{P}(\mathbf{c}_{N/2})$, is a non empty close interval, so that we can define:

$$\mathrm{c}^-_{(N+1)/2}(\mathbf{c}_{N/2}) = \min_{\mu\in\mathrm{P}(\mathbf{c}_{N/2})} \{\mathrm{c}_{(N+1)/2}(\mu)\}, \quad \mathrm{c}^+_{(N+1)/2}(\mathbf{c}_{N/2}) = \max_{\mu\in\mathrm{P}(\mathbf{c}_{N/2}} \{\mathrm{c}_{(N+1)/2}(\mu)\}. \tag{4.27}$$

We define the canonical fractional moment as follows:

$$p_{k/2} = \frac{\mathrm{c}_{k/2} - \mathrm{c}^-_{k/2}(\mathbf{c}_{k/2})}{\mathrm{c}^+_{k/2}(\mathbf{c}_{k/2}) - \mathrm{c}^-_{k/2}(\mathbf{c}_{k/2})}. \tag{4.28}$$

The canonical moment vector belongs to $[0,1]^N$, which has a simpler topology compared to the moment space. By using the results obtained on the canonical integer moments Dette and Studden(1997), we can write the algebraic relation between fractional moments and their corresponding canonical moments, in the case of $N = 3$, by using the identification (4.25):

$$p_{1/2} = \frac{\mathrm{m}_{1/2}}{\mathrm{m}_{0/2}}, \quad p_{2/2} = \frac{\mathrm{m}_{0/2}\mathrm{m}_{1/2} - \mathrm{m}_{1/2}^2}{(\mathrm{m}_{0/2} - \mathrm{m}_{1/2})\mathrm{m}_{1/2})}, \quad p_{3/2} = \frac{(\mathrm{m}_{0/2} - \mathrm{m}_{1/2})(\mathrm{m}_{1/2}\mathrm{m}_{3/2} - \mathrm{m}_{2/2}^2)}{(\mathrm{m}_0\mathrm{m}_{2/2} - \mathrm{m}_{1/2}^2)(\mathrm{m}_{1/2} - \mathrm{m}_{2/2})}. \tag{4.29}$$

The lower and upper principal representation associated to the fractional moments: if the moment vector $\mathbf{c}_{N/2} \in \mathbb{M}_N^{1/2}$, the vector $\mathbf{c}^{\pm}_{(N+1)/2} = (\mathrm{c}_{0/2}, \dots, \mathrm{c}_{N/2}, \mathrm{c}^{\pm}_{(N+1)/2}(\mathbf{c}_{N/2}))$ belongs to the boundary of the moment space $\mathbb{M}_{N+1}^{1/2}$, and the measure set $\mathrm{P}_{N+1}(\mathbf{c}^{\pm}_{(N+1)/2}) = \{\mu_{\pm}\}$ has only one element. The measure μ_+ (resp μ_-) is called the upper (resp lower) principal representation. In the case of an integer moment vector $\widetilde{\mathbf{c}}_N$, it was shown in Dette and Studden(1997) that the lower and upper principal representation ($\widetilde{\mu}_+$ and $\widetilde{\mu}_-$) can be expressed as sum of $n_s \leq (N+1)/2$ weighted delta-Dirac functions (we count the abscissas in the interior $]0,1[$ as 1 and the ones at the extremities, 0 or 1, as 1/2):

$$\widetilde{\mu}_{\pm}(r) = \sum_{i=1}^{n_s} \widetilde{w}_i^{\pm} \delta_{r_i^{\pm}}(r), \tag{4.30}$$

we recall that the subscript $\widetilde{\bullet}$ is used for integer moments and its corresponding measure, which are related to the fractional moments according to the identification (4.25). The Product-Difference (PD) algorithm Gordon(1968) can be used to determine the weights and the abscissas of the lower principal representation depending on the moments $\widetilde{\mathbf{c}}_N$. In other words, this algorithm solves the following non-linear system:

$$\widetilde{\mathrm{c}}_k = \sum_{i=1}^{n_s} \widetilde{w}_i^- r_-^k, \quad k = 0, 1, \dots, N. \tag{4.31}$$

To sump up, the identification (4.25) allows us to relate the lower principal representation μ_- of some fractional moments, to the lower principal representation $\widetilde{\mu}_-$ of its associated integer moments as follows:

$$\mu_-(x) \quad = \quad \frac{1}{2\sqrt{x}} \widetilde{\mu}_-(\sqrt{x}). \tag{4.32}$$

Proposition 4.4.1 *Let r be a positive real number, then for all positive real x we have:*

$$\frac{\delta_r(\sqrt{x})}{2\sqrt{x}} = \delta_{r^2}(x). \tag{4.33}$$

Using the last proposition, we deduce that:

$$\mu_-(x) = \sum_{i=1}^{n_s} w_i^- \delta_{x_i^-}(x), \tag{4.34}$$

where $x_i^- = (r_i^-)^2$ and $w_i^- = \widetilde{w}_i^-$.
Thus, we can reuse the PD algorithm to determine the weights and abscissas of the lower principal representation for fractional moments.

4.4.4 Maximum Entropy reconstruction

NDF reconstruction through the Maximum Entropy (ME) formalism provides a smooth reconstruction to close the moments system (4.24) as it was done in the EMSM model. The ME reconstruction consists in maximizing the Shannon entropy defined in (4.11), under the constraints that the first $N+1$ (in our case $N=3$) fractional moments of the size distribution are equal to the real moments

$$\mathrm{m}_{k/2} = \int_0^1 S^{k/2} n(S) dS, \quad k = 0 \dots N. \tag{4.35}$$

4.4.4.1 Existence and uniqueness of the solution

In this part, we give a proof of the existence and uniqueness of the ME distribution function. We mention that the case of the integer moments has been already treated in Mead and Papanicolaou(1984). We have used some of its ideas. But the present proof is completely different and simplified. Indeed, Mead and Papanicolaou(1984) have used the monotonic properties of the moments, which is a characterization of the integer moments space, to prove the existence of the ME solution. In our case, we will only use the definition of the fractional moments space.

The ME problem reads:

$$\max\left\{H[n]=-\int_0^1 n(S)\ln(n(S))dS\right\}, \quad \mathrm{m}_{k/2}=\int_0^1 S^{k/2}n(S)dS, \quad k=0\dots N, \tag{4.36}$$

where $\mathbf{m}_{N/2} = (\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, .., \mathrm{m}_{N/2})$ is a moment vector in the interior of the fractional moment space $\mathbb{M}_N^{1/2}([0,1])$.

Lemma 4.4.2 *If the constrained optimization problem* (4.36) *admits a solution, then this solution is unique and can be written in the following form:*

$$n^{ME}(S) = \exp(-\lambda_0 - \sum_{i=1}^{N} \lambda_i S^{i/2}), \tag{4.37}$$

where $\boldsymbol{\lambda} = (\lambda_0, .., \lambda_N)^t \in \mathbb{R}^N$. *Moreover, this problem is equivalent to finding an extremum of the potential function:*

$$G(\lambda_0, \dots, \lambda_N) = \int_0^1 \exp(-\lambda_0 - \sum_{i=1}^{N} \lambda_i S^{i/2}) dS + \sum_{k=0}^{N-1} \lambda_k \mathrm{m}_{k/2}. \tag{4.38}$$

Proof: The Lagrangian function associated to this standard constrained optimization problem is:

$$L(n,\boldsymbol{\lambda}) = H[n] - (\lambda_0 - 1)\left(\int_0^1 n(s)ds - \mathrm{m}_0\right) - \sum_{i=1}^{N}\left[\lambda_i\left(\int_0^1 s^{i/2}n(s)ds - \mathrm{m}_{i/2}\right)\right], \tag{4.39}$$

where $\boldsymbol{\lambda} = (\lambda_0, .., \lambda_N)$ is the vector of the Lagrange's multipliers.
Let us suppose that, for a given moment vector $\mathbf{m}_N$, there exists a number density function n^{ME}

which is the solution of the ME problem (4.36). Then, there exists a vector $\boldsymbol{\lambda}^{ME}$ for which the differential of the Lagrange function $L(n, \boldsymbol{\lambda})$ vanishes at the point $(n^{ME}, \boldsymbol{\lambda}^{ME})$:

$$\left\{\begin{array}{rcl} DL(n^{ME}, \boldsymbol{\lambda}) \cdot (h, \mathrm{d}\boldsymbol{\lambda} = \mathbf{0}) & = & \int_0^1 h(s) \left[-ln(n^{ME}(s)) - \sum\limits_{i=0}^{N} \lambda_i s^{i/2} \right] ds = 0, \\ \frac{\partial L}{\partial \lambda_i}(n^{ME}, \boldsymbol{\lambda}) & = & \int_0^1 s^{i/2} n^{ME}(s) ds - \mathrm{m}_{i/2} = 0, \end{array}\right. \tag{4.40}$$

where h is a positive function and $DL(n^{ME}, \boldsymbol{\lambda})\cdot$ is the differential of the Lagrange function at the point $(n^{ME}, \boldsymbol{\lambda}^{ME})$. Since the system (4.40) is valid for all h, it yields:

$$\left\{\begin{array}{rcl} n^{ME}(S) & = & \exp(-\lambda_0 - \sum\limits_{i=1}^{N} \lambda_i S^{i/2}), \\ \mathrm{m}_{k/2} & = & \int_0^1 s^{k/2} \exp(-\lambda_0 - \sum\limits_{i=1}^{N} \lambda_i s^{i/2}) ds. \end{array}\right. \tag{4.41}$$

The problem then consists in finding a vector $\boldsymbol{\lambda} = (\lambda_0, .., \lambda_N)$ in $\mathbb{R}^N$ that satisfies the moment equations in system (4.41). This problem is equivalent to find an optimum of the potential function $G(\lambda_0, .., \lambda_N)$ defined in (4.38).

The Hessian matrix H defined by $H_{i,j} = \frac{\partial^2 G}{\partial \lambda_i \partial \lambda_j}$ is a positive definite matrix, which ensures uniqueness of an eventual existing solution. □

Let us then study the function G.

Lemma 4.4.3 *The function G, defined in (4.38), is a continuous function in $\mathbb{R}^N$, and goes to infinity when $||\boldsymbol{\lambda}|| \to +\infty$.*

Proof: Let us suppose that the last assertion is wrong, so there exists a sequence $(\boldsymbol{\lambda}^{(n)})_{n=0,1,..}$ such that $||\boldsymbol{\lambda}^{(n)}|| \to +\infty$ when $n \to +\infty$ and $\sup\limits_n \left\{ G(\boldsymbol{\lambda}^{(n)}) \right\} < +\infty$.
Hence, there exists $A \in \mathbb{R}$ such that:

$$G(\boldsymbol{\lambda}^{(n)}) = \int_0^1 \exp(-\sum_{i=0}^{N} \lambda_i^{(n)} S^{i/2}) dS + \sum_{k=0}^{N} \lambda_k^{(n)} \mathrm{m}_{k/2} < A. \tag{4.42}$$

We write for each $n \in \mathbb{N}$, $\boldsymbol{\lambda}^{(n)} = \lambda^n(\alpha_0^{(n)}, \alpha_0^{(n)}, \ldots, \alpha_N^{(n)})$, such that $\sum_{i=0}^{N} (\alpha_i^{(n)})^2 = 1$ and $\lambda^{(n)} \to +\infty$.
Since the sequence $(\boldsymbol{\alpha}^{(n)})_{n=0,1..}$ is a bounded sequence, we can subtract a convergent subsequence $(\boldsymbol{\alpha}^{\phi(n)})_n$, where $\phi : \mathbb{N} \to \mathbb{N}$ is an increasing function and:

$$\lim_{n \to +\infty} \alpha_i^{\phi(n)} = \alpha_i. \tag{4.43}$$

To simplify the notation, we can directly consider that $\alpha_i^{(n)} \to \alpha_i$ when $n \to +\infty$.
We note by $Q^{(n)}(x) = \sum_{i=0}^{N} (\alpha_i^{(n)} x^{i/2})$ and $Q(x) = \sum_{i=0}^{N} \alpha_i x^{i/2}$.

Since the vector $\mathbf{m}_{N/2} = (\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, .., \mathrm{m}_{N/2})$ is a moment vector, there exists a non-negative distribution function f such that $m_{k/2} = \int_0^1 s^{k/2} f(s) ds$ for $k = 0, \ldots N$, and

$$G(\boldsymbol{\lambda}^{(n)}) = \int_0^1 \exp(-\lambda^{(n)} Q^{(n)}(s)) ds + \int_0^1 \lambda^{(n)} Q^{(n)}(s) f(s) ds \leq A. \tag{4.44}$$

Since the first integral is positive

$$\int_0^1 Q^{(n)}(s)f(s)ds \le \frac{A}{\lambda^{(n)}}. \tag{4.45}$$

When n tends to infinity, we get:

$$\int_0^1 Q(s)f(s)ds \le 0. \tag{4.46}$$

We have $Q \neq 0$, $f \geq 0$ and $f \neq 0$, and since Q is a continuous function, it follows from the inequality (4.46) that there exists $[a, b] \subset [0, 1]$ in which $Q(s) \leq -B$ and $B > 0$. Since $Q^{(n)}$ converges uniformly to Q in $[0, 1]$, then, for all $s \in [a, b]$ and for the large enough values of n:

$$Q^{(n)}(s) < -B/2. \tag{4.47}$$

Using these results in the inequality (4.44), as well as $\alpha_k^n \mathrm{m}_{k/2} \geq -\mathrm{m}_0$, we get:

$$\begin{aligned} A &\geq \textstyle\int_a^b (exp(-\lambda^{(n)} Q^{(n)}(s)))ds + \sum_{i=0}^N \lambda_i^{(n)} \mathrm{m}_{i/2}, \\ &\geq (b-a)exp(\lambda^{(n)}(\tfrac{B}{2})) - \lambda^{(n)} N \mathrm{m}_0, \end{aligned} \tag{4.48}$$

In the limit when n goes to infinity, we get the contradiction $+\infty < A$, thus concluding the proof. □

Theorem 4.4.4 *If the vector* $\mathbf{m}_{N/2} = (\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, .., \mathrm{m}_{N/2})$ *belongs to the interior of the Nth fractional moment space, then the constrained optimization* (4.36) *problem admits a unique solution, which is in the following form:*

$$n^{ME}(S) = \exp(-\lambda_0 - \sum_{i=1}^N \lambda_i S^{i/2}). \tag{4.49}$$

Proof: The proof is straightforward by using the two last lemmas. □

4.4.5 Algorithm of the NDF reconstruction through the Entropy Maximization

The reconstruction of the NDF through the maximization of the Shannon entropy goes back to finding the Lagrange's multipliers $\lambda_0 \ldots \lambda_N$ such that:

$$\mathrm{m}_{k/2} = \int_0^1 S^{k/2} \exp(-(\lambda_0 + \sum_{i=1}^N \lambda_i S^{i/2})dS, \tag{4.50}$$

where $k = 0, \ldots, N$. Solving this nonlinear system is equivalent to minimizing the convex function $G(\boldsymbol{\lambda})$. We solve the problem by using Newton-Raphson iteration as proposed in Mead and Papanicolaou(1984). The ME reconstruction is then summed up in Algorithm 1.

The integral computations are done by using Gauss-Legendre quadrature. In Mead and Papanicolaou(1984), it is shown that 24-point quadrature is very efficient to calculate accurately the different integral expressions involved in Algorithm 1.

Algorithm 1 ME algorithm

```
Choose initial guess of the vector $\boldsymbol{\lambda}$.
$\delta_{k/2} \leftarrow \mathrm{m}_{k/2} - \int_0^1 S^{k/2} \exp(-\sum_{i=0}^{N} \lambda_i S^{i/2})$
while $||\boldsymbol{\delta}|| > \epsilon \mathrm{m}_0$ do
    for $i, j < N$ do
        $H_{i,j} \leftarrow \int_0^1 S^{(i+j)/2} \exp(-\sum_{i=0}^{N} \lambda_i S^{i/2})$
    end for
    $\boldsymbol{\lambda} \leftarrow \boldsymbol{\lambda} - \boldsymbol{H}^{-1} \cdot \boldsymbol{\delta}$
    for $k < N$ do
        $\delta_{k/2} \leftarrow \mathrm{m}_{k/2} - \int_0^1 S^{k/2} \exp(-\sum_{i=0}^{N} \lambda_i S^{i/2})$
    end for
end while
```

4.5 Conclusion

In this chapter, we have proposed a new model for an evaporating polydisperse spray with the capacity of describing the interface geometry in the disperse phase. This description is established by analogy with the interface description in interfacial flow. The present model is a first step toward a coupling with a two-fluid model, where additional transport equations of geometrical variables are used. In this way, these information allow to describe the atomization process and to capture the polydispersion in the disperse phase. The present model, dedicated to the simulation of an evaporating spray, involves high order fractional size-moments, where the size is the droplet area. The fractional moments space and its properties have been discussed during this chapter. They will be used in chapters 7 and 8 to develop robust, accurate and realizable numerical schemes. Furthermore, we have extended the continuous reconstruction through entropy maximization of integer moments Massot et al.(2010) to the case of fractional moments. Finally, we emphasize that this model can be extended to the case of a two-way coupling and to an anisotropic Gaussian closure in velocity for a more accurate modeling of the disperse phase features. However, the main challenge for the continuity of this work will lie in the modeling of the averaged geometrical variables evolution in the separated phases zone, where the key atomization mechanisms generating the polydisperse sprays are to be found.

Chapter 5

Statistical modeling of the gas-liquid interface

5.1 Introduction

In the previous chapter, we discussed how to enrich the gas-liquid interface description by using new geometrical information: the mean and Gauss curvatures, the surface area density and the volume fraction. These four variables have been related to high-order fractional size-moments model for sprays of spherical droplets and can be then used to describe the polydispersion of the droplets. In the present chapter, we continue at investigating the possibility to describe the gas-liquid interfaces by using geometrical information. First, inspired by the pioneering works of Pope(1988) for the description of the dynamics of flames, we define a Surface Density Function (SDF) within a different phase space. In our work, the phase space of this distribution is composed of the mean and Gauss curvatures and the interfacial velocity. The key issue is to make the link between the statistics of a local description of the interface and the statistical description of isolated objects through a number density function in an appropriate phase space. A discrete formalism of the SDF is then proposed to describe disperse phase flows. In this case, the geometrical quantities are averaged on the surface of isolated droplets or bubbles. We show that the system of equations of the moments of the discrete SDF can degenerate to the fractional moments model, that we have proposed in the previous chapter for sprays of spherical droplets. Going forward into a full generalization of the interface description, we define a spatially Averaged SDF (ASDF). This is based on an averaging kernel bounded to a small region around the interface. When performed in a consistent way, this process preserves some necessary topological properties of the gas-liquid interface description. This leads us to the definition of a Generalized NDF (GNDF), which can be used for both separated and disperse phases and which is shown to degenerate into a standard NDF in the disperse region. In the last section, we finally propose a new numerical procedure for the computation of the curvatures and of the different statistical distributions (SDF or NDF, based on a characteristic spatial averaging size) from the values of a Level Set function. These new algorithms are eventually applied to the post-processing of some DNS simulations, obtained with the ARCHER code, developed at CORIA laboratory Menard et al.(2007). The meaning of the obtained numerical results is two fold: they assess the theoretical part of this work by proving that we can indeed turn an interfacial information into a statistics of countable objects, but they also pave the way to a new way of analyzing DNS of interfacial flows.

5.2 Surface element properties and probabilistic description of the gas-liquid interface

Immiscible two-phase flows, such as gas-liquid mixtures, are characterized by the presence of a sharp interface. Indeed, the gas-liquid interface thickness being of the order of the molecular mean free path ($\lambda = 10$ nm), it is smaller than the microscopic length scales of the vast majority of two-phase flow applications. Thus, at macroscopic scale, this interface can be represented as a 2D dynamic surface embedded in a 3D domain. In the following, we consider a two-phase flow (gas and liquid) within a finite 3D domain Ω_x. Let us denote by $\Sigma_I(t)$ the moving surface that separates both phases. First, we do not make any assumption on the flow topology (separated or disperse). We start by defining a 2D surface and some intrinsic geometrical variables in subsections 5.2.1-5.2.3. Then in subsection 5.2.4, we introduce a statistical description of the gas-liquid interface similar to Pope's description of flames dynamics and propagation Pope(1988).

5.2.1 Surface definition

Since we are dealing with disperse and separated phases in the same domain, the moving surface $\Sigma_I(t)$ is not necessarily a connected set: in general it consists in a set of closed (disperse phase) and unclosed (bulk phase) connected sub-surfaces. Therefore, the global surface $\Sigma_I(t)$ can be written as a union of connected sub-surfaces $\Sigma_i(t)$:

$$\Sigma_I(t) = \bigcup_{i=1}^{N_{\max}} \Sigma_i(t), \tag{5.1}$$

that are separated in the sense that, for $i \neq j$:

$$\min_{(\boldsymbol{x},\boldsymbol{y})\in\Sigma_i(t)\times\Sigma_j(t)} (||\boldsymbol{x}-\boldsymbol{y}||) > 0.$$

Now, let us consider two standard descriptions of a moving surface. First, the surface can be defined as the set of zeros of a time-space function $(t;\boldsymbol{x}) \mapsto g(t;\boldsymbol{x})$:

$$\Sigma_I(t) = \{\boldsymbol{x} \in \Omega_x; \quad g(t;\boldsymbol{x}) = 0\}. \tag{5.2}$$

The function g satisfies the following kinematic equation:

$$\frac{\partial g}{\partial t} + \mathbf{v_I}^g \cdot \boldsymbol{\nabla}(g) = 0, \tag{5.3}$$

where $\mathbf{v_I}^g(t;\boldsymbol{x})$ is an interface velocity, whose definition is discussed thereafter. Let us call this approach: the "implicit definition of a surface".

On the other hand, the "explicit definition of a surface", also called the "parametric surface", consists in using two real parameters u, v and parametric functions:

$$\boldsymbol{X}_I(t;.) : \begin{cases} \mathcal{U}_i & \longrightarrow & \Omega_x \\ (u,v) & \longmapsto & \boldsymbol{X}_I(t;u,v) \end{cases}, \quad i \in \{1,\dots,N_{\max}\}, \tag{5.4}$$

where $\mathcal{U}_i$ is a subset of $\mathbb{R}^2$, such that:

$$\Sigma_i(t) = \{\boldsymbol{X}_I(t;u,v); \quad (u,v) \in \mathcal{U}_i\}. \tag{5.5}$$

Each couple of parameters (u, v) is associated to a point located on the surface and moving according to an interface velocity that depends on the choice of the parametrization:

$$\frac{\partial \mathbf{X}_I(t; u, v)}{\partial t} = \mathbf{v}_I{}^{\mathcal{U}}(t, \mathbf{X}_I(t; u, v)). \tag{5.6}$$

We will discuss in the next section the relation between the interface velocities used in the implicit definition $\mathbf{v}_I{}^g$ and in the explicit definition $\mathbf{v}_I{}^{\mathcal{U}}$.

Although we mostly consider the implicit definition of the surface (5.2), some of the definitions provided in the following will also be given for a parametric description of the surface for the sake of clarity. From now on, we suppose that each sub-surface Σ_i is a C^2 oriented surface, meaning that the space function $g(t; .)$ is a C^2 differentiable function. The orientation is chosen such that the gradient $\nabla g(t; \boldsymbol{x})$ at the interface points $(g(t; \boldsymbol{x}) = 0)$ is strictly oriented toward the gaseous phase.

5.2.2 Intrinsic gas-liquid interface variables

The aim of this paragraph is to introduce some local intrinsic properties of the interface, meaning that the quantities associated with these properties do not depend on the way the surface is defined (implicitly or explicitly, choice of a parametrization...). These quantities will be useful for setting a statistical description of the interface.

5.2.2.1 Normal vector and tangent plane

Let us consider a point on the surface $\boldsymbol{x}_i \in \Sigma_I(t)$. Using (5.2), the normal vector at $\boldsymbol{x}_i$ is given by:

$$\boldsymbol{n}(t, \boldsymbol{x}_i) = \frac{\nabla_{\boldsymbol{x}}(g)(t; \boldsymbol{x}_i)}{||\nabla_{\boldsymbol{x}}(g)(t; \boldsymbol{x}_i)||}. \tag{5.7}$$

The tangent plane to the surface at $\boldsymbol{x}_i$ is the unique plan orthogonal to $\boldsymbol{n}$ and passing by $\boldsymbol{x}_i$. One can consider that the tangent plane provides a first order approximation of the surface at $\boldsymbol{x}_i$.

5.2.2.2 Curvatures

Curvatures are defined as the infinitesimal variations of $(t, \boldsymbol{x}) \mapsto \boldsymbol{n}(t, \boldsymbol{x})$ when $\boldsymbol{x}$ follows a path over the interface. These variations can be expressed as a function of the Hessian matrix $\mathcal{H}(g)$ of the function $g(t; .)$ according to (see also Kindlmann et al.(2003)):

$$\nabla_{\boldsymbol{x}}(\boldsymbol{n})^T = -\frac{1}{||\nabla_{\boldsymbol{x}}(g)||}(\mathbf{I}_3 - \boldsymbol{n} \otimes \boldsymbol{n}^T)\mathcal{H}(g), \tag{5.8}$$

where $\mathcal{H}(g)$ is given by:

$$\mathcal{H}(g)(t; \boldsymbol{x}) = \nabla_{\boldsymbol{x}}((\nabla_{\boldsymbol{x}}(g))) = \left(\frac{\partial^2 g}{\partial x_j \partial x_k}(t; \boldsymbol{x})\right)_{j,k=1...3}, \tag{5.9}$$

and $\mathbf{I}_3$ is the identity matrix.

It can be shown that there exists an orthonormal basis $\{\boldsymbol{e}_1^*, \boldsymbol{e}_2^*\}$ of the tangent plane at the surface point $\boldsymbol{x}_i$, such that the restriction of the matrix $\boldsymbol{\nabla}_{\boldsymbol{x}}(\boldsymbol{n})^T$ to the tangent plane is a 2×2 diagonal matrix Kindlmann et al.(2003). In other words, in the orthonormal basis $\{\boldsymbol{e}_1^*, \boldsymbol{e}_2^*, \boldsymbol{n}\}$, $\boldsymbol{\nabla}_{\boldsymbol{x}}(\boldsymbol{n})^T$ reads:

$$\boldsymbol{\nabla}_{\boldsymbol{x}}(\boldsymbol{n})^T = \begin{pmatrix} \kappa_1 & 0 & \sigma_1 \\ 0 & \kappa_2 & \sigma_2 \\ 0 & 0 & 0 \end{pmatrix}, \tag{5.10}$$

where $\kappa_1 \geq \kappa_2$ are the two principal curvatures and (σ_1, σ_2) are two real variables. The eigenvectors $\{\boldsymbol{e}_1^*, \boldsymbol{e}_2^*\}$ corresponding to the eigenvalues (κ_1, κ_2) are also called the *principal directions* of the surface at $\boldsymbol{x}_i$.

Instead of using the two principal curvatures, one can consider the *mean curvature* H and the *Gauss curvature* G, defined by:

$$\begin{aligned} H &= \frac{1}{2}(\kappa_1 + \kappa_2), \\ G &= \kappa_1 \kappa_2. \end{aligned} \tag{5.11}$$

Indeed, the mapping

$$\left\{ \begin{array}{ccc} \{(\kappa_1, \kappa_2) \in \mathbb{R}^2;\ \kappa_1 \geq \kappa_2\} & \longrightarrow & \left\{(H, G) \in \mathbb{R}_+^2;\ H^2 \geq G\right\} \\ (\kappa_1, \kappa_2) & \longmapsto & (H, G) \end{array} \right., \tag{5.12}$$

is one-to-one.

5.2.2.3 Area density measure and stretch rate

The last quantity needed in the following is an evaluation of the interface area within any control volume. For this purpose, one defines the *area density measure* as follows, see Morel(2015) and related works:

$$\delta_I(t; \boldsymbol{x}) = ||\boldsymbol{\nabla}_{\boldsymbol{x}}(g)||\, \delta(g(t, \boldsymbol{x})), \tag{5.13}$$

where δ denotes the Dirac measure. Consequently, for any volume $\mathcal{V}$, the area $A_{\mathcal{V}}(t)$ of surface contained in $\mathcal{V}$ at time t is given by:

$$A_{\mathcal{V}}(t) = \int_{\Omega_x} \mathbb{1}_{\mathcal{V}}(\boldsymbol{x}) \delta_I(t; \boldsymbol{x}) d\boldsymbol{x},$$

where $\mathbb{1}_{\mathcal{V}}(\boldsymbol{x})$ is the characteristic function of volume $\mathcal{V}$. It is important to note that the measure δ_I does not depend on the choice of g. It is thus an intrinsic property of the interface.

5.2.2.4 Interface velocity

We have seen in paragraph 5.2.1 that the interface velocity may have multiple definitions, according to the definition of the interface (implicit (5.2) or explicit (5.4)) or the choice of the parametrization. Let us choose here a unique definition of this velocity.

As underlined in Drew(1990), one can see that the evolution of the interface according to (5.3) only depends on the normal component of $\mathbf{v}_I{}^g$. Let us then introduce $\mathrm{v}_I = \mathbf{v}_I{}^g \cdot \boldsymbol{n}$, that is unambiguously defined, and the interface velocity by:

$$\mathbf{v}_I = \mathrm{v}_I \boldsymbol{n}, \tag{5.14}$$

By considering a point $\boldsymbol{X}_I(t;u,v)$ on the surface for a given parametrization and the application $\varphi : (t;u,v) \mapsto g(t;\boldsymbol{X}_I(t;u,v))$, one can show the following relation with the the velocity ${\mathbf{v}_I}^{\mathcal{U}}$ defined in (5.6):

$$\mathrm{v}_I = {\mathbf{v}_I}^{\mathcal{U}} \cdot \boldsymbol{n}.$$

Now, let us consider that the interface lies in a medium, whose velocity is $\mathbf{U}(t,\boldsymbol{x})$. Moreover, this interface may propagate, normal to itself, at a velocity $V_e\boldsymbol{n}$ relative to the medium Pope(1988). This is for example the case of a flame that separates burnt and unburnt gases, for which the flame speed V_e depends on the chemical reactions rates. Or in the case of two-phase flows, V_e may characterize the rate of phase transitions, such as evaporation. The interfacial velocity is then related to these quantities through:

$$\mathrm{v}_I = (\mathbf{U} \cdot \boldsymbol{n}) + V_e.$$

5.2.3 Time evolution of interfacial variables

In this paragraph, we present the evolution equations for surface element properties based on the works of Drew(1990). Let us mention that the choice in the definition of the interface velocity affects the expression of the evolution laws for the curvatures and the area density measure. Here we consider the interface velocity defined by (5.14). Then, the time evolution of the curvatures can be expressed as follows:

$$\begin{aligned} \dot{H} &= -\frac{1}{2}\nabla_T^2(\mathrm{v}_I) - (2H^2 - G)\mathrm{v}_I, \\ \dot{G} &= -H\nabla_T^2(\mathrm{v}_I) + \sqrt{H^2 - G}\left(\frac{\partial^2 \mathrm{v}_I}{\partial {y_1^*}^2} - \frac{\partial^2 \mathrm{v}_I}{\partial {y_2^*}^2}\right) - 2\,H\,G\,\mathrm{v}_I, \end{aligned} \tag{5.15}$$

where $\dot{\bullet} = \partial_t \bullet + \mathbf{v}_I \cdot \nabla_{\boldsymbol{x}} \bullet$, denotes the Lagrangian time derivative, $\nabla_T^2 = \partial_{y_1}^2 + \partial_{y_2}^2$ is the tangential Laplacian operator (y_1 and y_2 can be any two orthonormal directions tangential to the surface), and y_k^* is the coordinate along the principal directions $\boldsymbol{e}_k^*$, $k = 1,2$. In B, we derive the equation for δ_I. This equation reads:

$$\dot{\delta}_I = -\left(\nabla_{\boldsymbol{x}} \cdot \mathbf{v}_I\right)\delta_I + 2H\,\mathrm{v}_I\,\delta_I. \tag{5.16}$$

In equation (5.16), one can identify the second term in the right-hand side with the *stretching rate* $\dot{\mathcal{S}}$, which is defined in Pope(1988) as an intrinsic property. For an interface velocity that is normal to the interface, its equation reads:

$$\dot{\mathcal{S}} = 2H\,\mathrm{v}_I. \tag{5.17}$$

Another important point in the comparison with Pope's equations is to note that in Pope(1988), the interface velocity is defined as the sum of the fluid velocity and the propagation velocity of the flame interface due to chemical reaction (combustion). Consequently, this leads to different equations for $\dot{H}$, $\dot{G}$ and $\dot{\mathcal{S}}$.

The system (5.15) is not closed, because of the second order derivatives of the interfacial velocity in the two principal directions. For future works, these terms need to be modeled, but this is not the objective of the present work. An interpretation of the different terms occurring in the system of equation (5.15) can be found in Drew(1990).

5.2.4 General statistical description of the interface

Now, we propose to introduce a general statistical description of a gas-liquid interface, that may be useful when the exact location of the interface is not known, like in turbulent flows Pope(1988) or in two-phase transition zones. Obviously, the statistical description of the interface can not be restricted to one geometrical variable, as is often done for disperse phases. Indeed, when assuming that droplets are spherical, the information about the droplets radii is enough to describe the interface geometry. This is however not the case with a general interface, such as one described by (5.1). In the following, we propose to use the mean and Gauss curvatures, as well as the interface velocity, to characterize the local interface properties and their evolution. These variables will be called the internal or phase space variables:

$$\widehat{\boldsymbol{\xi}} = (\widehat{H}, \widehat{G}, \widehat{\mathbf{v}_I}),$$

the set of values that can be attained by any realization $\widehat{\boldsymbol{\xi}} = \boldsymbol{\xi}(t, \boldsymbol{x})$ at time t and position $\boldsymbol{x}$ is the phase space Ω_{ξ}. Moreover we need to define an appropriate probabilistic measure. When the interface splits the two phases into a certain number of discrete and countable particles, a relevant measure is the NDF. In the case of a general gas-liquid interface, we adopt a measure based on the area concentration of the geometric properties of the interface. Let us consider the surface density function (SDF) $F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}})$ Pope(1988), defined as follows: the quantity $F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}}) d^5\widehat{\boldsymbol{\xi}} d^3\boldsymbol{x}$ measures the probable surface area present in the spatial volume $d^3\boldsymbol{x}$ around $\boldsymbol{x}$ and in the phase-space volume $d^5\widehat{\boldsymbol{\xi}}$ around $\widehat{\boldsymbol{\xi}}$. The notion "probable" is used here in the sense of an ensemble averaging over different realizations, as defined by Drew Drew and Passman(1999). To clarify this concept of SDF, let us first give a definition of the SDF for one realization, namely the fine-grain SDF, $F^{'}$, using the implicit definition of the surface (5.2):

$$F^{'}(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}}) = \delta_I(t, \boldsymbol{x}) \delta(\widehat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, \boldsymbol{x})). \tag{5.18}$$

Pope also defines the fine-grain SDF in Pope(1988), using the explicit definition (5.4):

$$F^{'}(t, \boldsymbol{x}, \widehat{\boldsymbol{\xi}}) = \int_{\mathcal{U}} L(t, u, v; \widehat{\boldsymbol{\xi}}) A(t, u, v) du dv, \tag{5.19}$$

where $A(t, u, v)$ is the area of a surface element, defined by:

$$A(t; u, v) = ||\partial_u \mathbf{X}_I \times \partial_v \mathbf{X}_I||(t; u, v) \tag{5.20}$$

and

$$L(t, u, v; \widehat{\boldsymbol{\xi}}) = \delta(\boldsymbol{x} - \boldsymbol{x}(t, u, v)) \delta(\widehat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, u, v)). \tag{5.21}$$

One can show that the two definitions are equivalent and independent of the choice of the parametrization and the function g.

Secondly, the SDF is defined as an ensemble average over all the realizations:

$$F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}}) =< F^{'}(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}}) >, \tag{5.22}$$

where $< \bullet >$ is the ensemble average operator.

Using the same procedure as in Pope(1988), we can derive the following transport equation with a source term:

$$\partial_t F + \nabla_x \cdot \{\widehat{\mathbf{v}_I} F\} + \nabla_{\widehat{\xi}} \cdot \{< \dot{\boldsymbol{\xi}} >_c F\} =< \dot{\mathcal{S}} >_c F, \tag{5.23}$$

where the *conditional expectation* of a scalar ψ is defined by:

$$< \psi >_c = \frac{< \psi F^{'}(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}}) >}{F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}})}.$$

The SDF evolves due to two main contributions: the left-hand side of equation (5.23) contributes to the evolution of F in the phase-space Ω_ξ and in the physical space Ω_x. This evolution is expressed as divergences of conservative fluxes. The source terms of the right-hand side of equation (5.23) express the evolution of the surface area due to stretching.

The conditional expectations of $\dot{H}$, $\dot{G}$ and $\dot{\mathcal{S}}$ may be obtained from their Lagrangian time evolution (5.15) by applying the ensemble averaging and by using the linearity of the average operator, which reads (λ is a constant):

$$< \lambda a + b > = \lambda < a > + < b >,$$

as well as Gauss and Leibniz rules:

$$< \partial_t a > = \partial_t < a > \quad \text{and} \quad < \partial_x a > = \partial_x < a > .$$

Then, the averaged evolution equations read:

$$\begin{aligned}
< \dot{H} >_c &= -\frac{1}{2} < \nabla_T^2(\mathrm{v}_I) >_c - (2\widehat{H}^2 - \widehat{G})\,(\widehat{\mathbf{v}}_I \cdot < \boldsymbol{n} >_c), \\
< \dot{G} >_c &= -\widehat{H} < \nabla_T^2(\mathrm{v}_I) >_c + \sqrt{\widehat{H}^2 - \widehat{G}} \left\langle \frac{\partial^2 \mathrm{v}_I}{\partial y_1^{*2}} - \frac{\partial^2 \mathrm{v}_I}{\partial y_2^{*2}} \right\rangle_c - 2\widehat{H}\widehat{G}\,(\widehat{\mathbf{v}}_I \cdot < \boldsymbol{n} >_c), \\
< \dot{\mathcal{S}} >_c &= 2\widehat{H}\,(\widehat{\mathbf{v}}_I \cdot < \boldsymbol{n} >_c).
\end{aligned} \tag{5.24}$$

One can note that some terms are unclosed in these equations. These are the conditional expectations of:

- the normal vector $< \boldsymbol{n} >_c$,
- the second order derivatives of the interfacial velocity $< \nabla_T^2(\mathrm{v}_I) >_c$,
- the evolution of the interfacial velocity $< \dot{\mathbf{v}}_I >_c$.

These terms need to be modeled and related to the internal flow dynamics of the gas and liquid phases.

5.2.5 Averaged quantities and moments of the SDF

The numerical resolution of equation (5.23) is unreachable for most applications because of the large dimension of the phase space Ω_ξ. In fact, solving the exact SDF would provide a level of detail on the flow, which is often not needed. Moreover, the large amount of data produced to attain this level of detail may hide the prevailing macroscopic features we are looking for. Therefore, we only aim at predicting some macroscopic variables of the flow, that ensure a satisfactory result for most industrial applications. Drew Drew(1990) derived Eulerian equations for the evolution of the following averaged quantities: the expected surface density $\Sigma(t, \boldsymbol{x})$, the interfacial-expected mean and Gauss curvatures, $\widetilde{H}$, $\widetilde{G}$ and the volume fraction $\alpha(t, \boldsymbol{x})$. In the present approach, we can not express the volume fraction as a function of the SDF without making a topological assumption on the gas-liquid interface. Therefore, in this part we temporarily restrict our study to the first three averaged interfacial quantities.

In terms of moments of the SDF, $\Sigma(t,\boldsymbol{x})$, $\widetilde{H}$ and $\widetilde{G}$ can be expressed as follows:

$$\begin{array}{rcl} \Sigma(t,\boldsymbol{x}) & = & M_{0,0,\mathbf{0}}(t,\boldsymbol{x}), \\ \Sigma(t,\boldsymbol{x})\widetilde{H} & = & M_{1,0,\mathbf{0}}(t,\boldsymbol{x}), \\ \Sigma(t,\boldsymbol{x})\widetilde{G} & = & M_{0,1,\mathbf{0}}(t,\boldsymbol{x}), \end{array} \tag{5.25}$$

where the moments read:

$$M_{i,j,\boldsymbol{l}}(t,\boldsymbol{x}) = \int_{\Omega_\xi} H^i G^j {V_{I,x}}^{l_x} {V_{I,y}}^{l_y} {V_{I,z}}^{l_z} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}) d^5\widehat{\boldsymbol{\xi}}, \tag{5.26}$$

and $\boldsymbol{l} = (l_x, l_y, l_z)$.

Then, by multiplying the SDF evolution equation (5.23) by $H^i G^j {V_{I,x}}^{l_x} {V_{I,y}}^{l_y} {V_{I,z}}^{l_z}$ and integrating over the whole phase-space domain, we can derive the evolution equations for the three quantities of interest:

$$\begin{array}{rcl} \partial_t \Sigma + \nabla_{\boldsymbol{x}} \cdot (\{\Sigma\, \overline{\mathbf{v}_I}\}) & = & \int_{\widehat{\xi}} < \dot{\mathcal{S}} >_c F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\, d^5\widehat{\boldsymbol{\xi}}, \\ \partial_t \Sigma\widetilde{H} + \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma\, \widetilde{H}\, \overline{\mathbf{v}_I}\right\}\right) & = & \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma\widetilde{H}(\overline{\mathbf{v}_I} - {\overline{\mathbf{v}_I}}^H)\right\}\right) + \int_{\widehat{\xi}} < \dot{\mathcal{S}} + \dot{H}/\widehat{H} >_c \widehat{H} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\, d^5\widehat{\boldsymbol{\xi}}, \\ \partial_t \Sigma\widetilde{G} + \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma\, \widetilde{G}\, \overline{\mathbf{v}_I}\right\}\right) & = & \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma\widetilde{G}(\overline{\mathbf{v}_I} - {\overline{\mathbf{v}_I}}^G)\right\}\right) + \int_{\widehat{\xi}} < \dot{\mathcal{S}} + \dot{G}/\widehat{G} >_c \widehat{G} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\, d^5\widehat{\boldsymbol{\xi}}. \end{array} \tag{5.27}$$

We can notice three types of averaged interfacial velocity that read:

$$\begin{array}{rcl} \Sigma\, \widetilde{\mathbf{v}}_I & = & \int_{\widehat{\xi}} \widehat{\mathbf{v}}_I F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\, d^5\widehat{\boldsymbol{\xi}}, \\ \Sigma\, \widetilde{H}\, {\overline{\mathbf{v}_I}}^H & = & \int_{\widehat{\xi}} \widehat{\mathbf{v}}_I \widehat{H} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\, d^5\widehat{\boldsymbol{\xi}}, \\ \Sigma\, \widetilde{G}\, {\overline{\mathbf{v}_I}}^G & = & \int_{\widehat{\xi}} \widehat{\mathbf{v}}_I \widehat{G} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\, d^5\widehat{\boldsymbol{\xi}}. \end{array} \tag{5.28}$$

Using the time evolution of both curvatures and of the stretch factor (5.15) in the system of equations (5.27), we can show that this system is equivalent to the system derived by Drew in Drew(1990). However, two closure issues need to be tackled at two different stages for future works. First, we need time evolution laws to close the conditional averages ($< \dot{H} >_c, < \dot{G} >_c, < \dot{\mathcal{S}} >_c$), as already mentioned for the transport equation of the SDF. Next, we need to propose a procedure to reconstruct the SDF from the known moments. Let us underline that, by using only the three first moments $M_{0,0,\mathbf{0}}$, $M_{1,0,\mathbf{0}}$, $M_{0,1,\mathbf{0}}$, we can not capture the variance of the phase-space coordinates. One should consider higher order moments. However, one must be careful when reconstructing the SDF in this case. Indeed, the curvatures $(\widehat{H}, \widehat{G})$ must satisfy $\widehat{H}^2 \geq \widehat{G}$ (see equation (5.12)), and the SDF reconstruction has to respect this constraint.

5.3 Probabilistic description of the gas-liquid interface for a disperse phase

5.3.1 Surface density function in the context of discrete particles

The SDF measures the pointwise probable area concentration for a probabilistic event, which is characterized by the phase space variables $\widehat{\boldsymbol{\xi}} = \{\widehat{H}, \widehat{G}, \widehat{\mathbf{v}}_I\} \in \Omega_\xi$ and evaluated at a local point

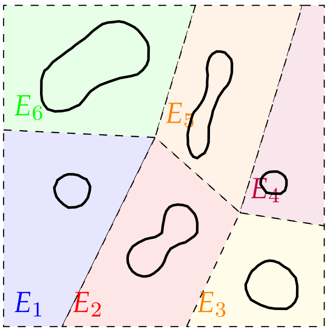


Figure 5.1: *Illustration of the spatial decomposition, with subspaces containing only one particle.*

$\boldsymbol{x} \in \Omega_x$. This probabilistic description is a pointwise description of the interface, which makes it general and valid for both separated and disperse phases. However, in the case of a disperse flow, a point-particle approach is usually considered. Each particle, droplet or bubble, is reduced to a point, characterized by some averaged and global quantities, such as: the velocity of the center of mass, the particle size (measured by volume, surface area or diameter), etc... and the statistics are conducted on a number of objects. In order to adapt the statistical approach presented in the previous section to a discrete formalism for a disperse phase, we first define interfacial quantities that are averaged over the surface of each particle. Then, we define a *discrete surface density function* (DSDF) that may characterize the dispersion in size and in shape of the population of particles.

Let us consider an isolated particle p_k of surface Σ_k, supposed to be smooth. Let us also consider a partition of the domain $\Omega_x = \bigcup\limits_{k=1}^{N_{max}} E_k$, such that each sub-domain E_k contains exactly one particle p_k, as illustrated in Figure 5.1. For a quantity $\phi(t, \boldsymbol{x})$, that can be a scalar (like curvatures) or a vector (like a velocity), we define its interface average $\overline{\phi}_k(t, \boldsymbol{x})$ over the particle surface Σ_k as follows:

$$\overline{\phi}_k(t, \boldsymbol{x}_k) = \frac{1}{S_k} \int_{\boldsymbol{x} \in E_k} \phi(t, \boldsymbol{x}) \delta_I(t, \boldsymbol{x}) d^3\boldsymbol{x}, \tag{5.29}$$

where S_k is the total surface area of the particle:

$$S_k = \int_{\boldsymbol{x} \in E_k} \delta_I(t, \boldsymbol{x}) d^3\boldsymbol{x},$$

and $\boldsymbol{x}_k$ its center of mass.

From now on, we consider each particle p_k as punctual, located at its center of mass $\boldsymbol{x}_k(t)$, having a surface area S_k and the averaged interface properties $\overline{\boldsymbol{\xi}}_k = \{\overline{H}_k, \overline{G}_k, \overline{\mathbf{v}_I}_k\}$. We define the DSDF by:

$$F^d(t, \boldsymbol{x}, \widehat{\boldsymbol{\xi}}) =< \sum_{k=1}^{N_{max}} S_k(t) \delta(\boldsymbol{x} - \boldsymbol{x}_k(t)) \delta(\widehat{\boldsymbol{\xi}} - \overline{\boldsymbol{\xi}}_k(t)) > \tag{5.30}$$

where N_{max} is the maximum number of particles in the domain Ω_x over all realizations. One can note that we have chosen to locate the particles at their centers of mass. However, the surface averaging procedure involves a surface barycenter defined by $\widetilde{\boldsymbol{x}}_k = 1/S_k \int_{\Sigma_k} \boldsymbol{x} dS(\boldsymbol{x})$. In order to cope with our choice, which is more practical as far as the particle dynamics is considered, we propose to express the mean interface velocity as:

$$\overline{\mathbf{v}_{I}}_{k}(t)=\frac{d\boldsymbol{x}_k(t)}{dt}+\tilde{V}_k(t),$$

where $\tilde{V}_k(t)$ is a fluctuation velocity, that stands for the particle shape deformations. In the following, this fluctuation velocity is neglected and we simply consider that the averaged interface velocity of a particle p_k reads:

$$\overline{\mathbf{v}_{I}}_{k}(t)=\frac{d\boldsymbol{x}_k(t)}{dt}. \tag{5.31}$$

The DSDF (5.30) verifies a similar transport equation as (5.23):

$$\partial_t F^d+\boldsymbol{\nabla}_{\boldsymbol{x}}\cdot\left(\widehat{\mathrm{v}}_I F^d\right)+\boldsymbol{\nabla}_{\widehat{\boldsymbol{\xi}}}\cdot\left(<\dot{\overline{\boldsymbol{\xi}}}>_c F^d\right)=<\dot{\overline{\mathcal{S}}}>_c F^d. \tag{5.32}$$

Once more, the time evolution of the averaged curvatures needs to be modeled in order to close the system of equations. This is work in progress and is out of the scope of the present paper.

The "localized" SDF defined in section 5.2.4 and the DSDF are two different functions. Nonetheless, they contain similar pieces of information about the gas-liquid interface properties. Indeed, both functions provide the probable surface area of the interface having some geometrical properties given by the phase-space variables. For the "localized" SDF, the phase-space variables are given by the localized interface properties at a surface point, while for the DSDF, we have considered interface properties averaged by object (particle). The link between the two functions can be seen through the first order moments of the two functions:

Proposition 5.3.1 *Given a subdomain $E\subset\Omega_x$, such that its border does not cut any particle, the integral over this subdomain of the zeroth and first order moments of the two functions are equal:*

$$\begin{aligned}\int_{\boldsymbol{x}\in E}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x} &= \int_{\boldsymbol{x}\in E}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}F^d(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x}\\ \int_{\boldsymbol{x}\in E}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}\widehat{\boldsymbol{\xi}}_l F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x} &= \int_{\boldsymbol{x}\in E}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}\widehat{\boldsymbol{\xi}}_l F^d(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x}\end{aligned} \tag{5.33}$$

where $\widehat{\boldsymbol{\xi}}_l\in\left\{\widehat{H},\widehat{G},\widehat{\mathrm{v}}_{Ix},\widehat{\mathrm{v}}_{Iy},\widehat{\mathrm{v}}_{Iz}\right\}$.

Proof: We illustrate this result for a moment of order one on G and zero on the other phase variables ($\widehat{\boldsymbol{\xi}}_l=\widehat{G}$):

$$\begin{aligned}\int_{\boldsymbol{x}\in E}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}\widehat{G}F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x} &= <\int_{\boldsymbol{x}\in E}G(t;\boldsymbol{x})\,\delta_I(\boldsymbol{x})\,d^3\boldsymbol{x}>\\ &= <\sum_k\int_{\boldsymbol{x}\in E_k\cap E}G(t;\boldsymbol{x})\,\delta_I(\boldsymbol{x})\,d^3\boldsymbol{x}>\\ &= <\sum_{\{k;\,\boldsymbol{x}_k(t)\in E\}}S_k\overline{G}_k>\\ &= <\sum_{\{k;\,\boldsymbol{x}_k(t)\in E\}}\int_{\boldsymbol{x}\in E_k}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}\widehat{G}F^d(t,\boldsymbol{x},\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x}>\\ &= \int_{\boldsymbol{x}\in E}\int_{\widehat{\boldsymbol{\xi}}\in\Omega_\xi}\widehat{G}F^d(t,\boldsymbol{x},\widehat{\boldsymbol{\xi}})\,d^5\widehat{\boldsymbol{\xi}}d^3\boldsymbol{x}.\end{aligned} \tag{5.34}$$

□

We recall that the zeroth and first-order moments are related to the expected surface density and the two curvatures $(\Sigma,\Sigma\widetilde{H},\Sigma\widetilde{G})$, defined in (5.25), and to the expected interface velocity, defined in (5.28). In this work, we give a great importance to these quantities because they contain some topological information about the gas-liquid interface, such as the number density of particles, as we will show in next section. we underline that any future definition of a new SDF should preserve these moments.

5.3.2 Link between the DSDF and a NDF: derivation of a Williams-Boltzmann-like equation

Let us now present the relation between the DSDF (5.30) and a number density function (NDF) for particles, using a geometrical property of the averaged Gauss curvature, known as the *Gauss-Bonnet formula.*

Let M be a 3D bounded object and $\mathbb{S}(M)$ its surrounding surface, supposed to be smooth. The Gauss-Bonnet formula applied to $\mathbb{S}(M)$ states that Abbena et al.(2006):

$$\int_{\boldsymbol{x}\in\mathbb{S}(M)} G(\boldsymbol{x})dS(\boldsymbol{x}) = 2\pi\,\chi(M), \tag{5.35}$$

where $\chi(M)$ is the Euler characteristic of the surface $\mathbb{S}(M)$ and is a topological invariant, meaning that two homeomorphic surfaces have the same Euler characteristic. In the following, we consider smooth particle surfaces, that are all homeomorphic to a sphere, the Euler characteristic of which is 2. Therefore, we can relate the averaged Gauss curvature over the surface of a particle p_k to its surface area by:

$$S_k = \frac{4\pi}{\overline{G}_k}. \tag{5.36}$$

Since $\dfrac{S_k\overline{G}_k}{4\pi} = 1$ for each particle, we are able to count the total number of particles, just by looking at the interface average of G. Then, we can derive a relation between the DSDF F^d and a Number Density Function (NDF) f. This relation is given by the following proposition.

Proposition 5.3.2 *When the two-phase flow is purely disperse, the distribution function* $\dfrac{\widehat{G}}{4\pi}F^d(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})$ *is equal to the NDF* $f(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})$ *for a phase space made of the averaged curvatures and interface velocity* $\widehat{\boldsymbol{\xi}} = \{\widehat{H},\widehat{G},\widehat{\mathbf{v}_I}\}$*:*

$$\frac{\widehat{G}}{4\pi}F^d(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}) = f(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}) \tag{5.37}$$

Proof:

$$\begin{aligned}
\frac{\widehat{G}}{4\pi}F^d(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}) &= <\sum_{k=1}^{N_{max}} S_k(t)\delta(\boldsymbol{x}-\boldsymbol{x}_k(t))\left\{\frac{\widehat{G}}{4\pi}\right\}\delta(\widehat{\boldsymbol{\xi}}-\overline{\boldsymbol{\xi}}_k(t))) >,\\
&= <\sum_{k=1}^{N_{max}} S_k(t)\delta(\boldsymbol{x}-\boldsymbol{x}_k(t))\left\{\frac{\overline{G}_k(t)}{4\pi}\right\}\delta(\widehat{\boldsymbol{\xi}}-\overline{\boldsymbol{\xi}}_k(t)) >,\\
&= <\sum_{k=1}^{N_{max}} \delta(\boldsymbol{x}-\boldsymbol{x}_k(t))\delta(\widehat{\boldsymbol{\xi}}-\overline{\boldsymbol{\xi}}_k(t)) >,\\
&= f(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}).
\end{aligned} \tag{5.38}$$

□

From the transport equation of the DSDF (5.32), we can derive a General Popolation Balance Equation (GPBE) for $f(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}})$, that is similar to the Williams-Boltzmann equation:

$$\partial_t f + \nabla_{\boldsymbol{x}} \cdot \{\widehat{\mathbf{v}}_I f\} + \nabla_{\widehat{\boldsymbol{\xi}}} \cdot \{< \dot{\overline{\boldsymbol{\xi}}} >_c f\} = \left(< \dot{\overline{\mathcal{S}}} >_c + < \frac{\dot{\overline{\mathrm{G}}}}{\overline{\mathrm{G}}} >_c \right) f. \tag{5.39}$$

The right-hand term behaves as a source term. We can relate this term to topological evolutions of the Euler characteristic of droplets due to fragmentation, coalescence, ects.

Let us now consider a set of droplets that do not break up, nor coalesce and which are all the time homeomorphic to spheres. Using the Gauss-Bonnet result (5.36), we can show that:

$$\dot{\overline{\mathcal{S}}}_k = -\frac{\dot{\overline{\mathrm{G}}}_k(t)}{\overline{\mathrm{G}}_k(t)}. \tag{5.40}$$

Therefore, without break-up nor coalescence, equation (5.39) reduces to:

$$\partial_t f + \nabla_{\boldsymbol{x}} \cdot \{\widehat{\mathbf{v}}_I f\} + \nabla_{\widehat{\boldsymbol{\xi}}} \cdot \{< \dot{\overline{\boldsymbol{\xi}}} >_c f\} = 0. \tag{5.41}$$

Equation (5.41) represents the time evolution of the particles NDF due to transport in physical space at the velocity of their center of mass, and the evolution of the internal variables, which represents the particle surface deformations and their acceleration by interaction with the carrier gas.

Now, we consider the volume of particles as an additional variable of the phase-space. Let us underline that adding the volume to the phase-space variables is specific to particles and can not be easily conducted within the general statistical approach for gas-liquid interface, as introduced in section 5.2.4. Furthermore, we also replace the averaged Gauss curvature by the surface area of the particles, since the two variables are related by (5.36). In the following, we denote the new phase-space variables by $\overline{\boldsymbol{\xi}}^v = \{\overline{H}, S, v, \overline{\mathbf{v}_I}\}$, where $S = 4\pi/\overline{G}$ is the surface area of the particles and v is their volume. A new NDF may be defined as follows:

$$f(t,\boldsymbol{x},\widehat{\overline{\boldsymbol{\xi}}^v}) =< \sum_{k=1}^{N_{max}} \delta(\boldsymbol{x} - \boldsymbol{x}_k(t))\delta(\widehat{\overline{\boldsymbol{\xi}}^v} - \overline{\boldsymbol{\xi}}^v_k(t)) > . \tag{5.42}$$

Next, we focus on the evolution of the expected-average terms of the gas-liquid interface: the interface density area $\Sigma(t,\boldsymbol{x})$, the mean and Gauss curvatures, $\widetilde{H}(t,\boldsymbol{x})$ and $\widetilde{G}(t,\boldsymbol{x})$, introduced in section 5.2.4 and the volume fraction $\alpha(t,\boldsymbol{x})$. In the current context, these terms can be expressed as first order moments of the NDF as follows:

$$\begin{array}{rcl} \alpha(t,\boldsymbol{x}) & = & M_{0,0,1,\mathbf{0}}(t,\boldsymbol{x}), \\ \Sigma(t,\boldsymbol{x}) & = & M_{0,1,0,\mathbf{0}}(t,\boldsymbol{x}), \\ \Sigma(t,\boldsymbol{x})\tilde{H} & = & M_{1,1,0,\mathbf{0}}(t,\boldsymbol{x}), \\ \Sigma(t,\boldsymbol{x})\tilde{\mathrm{G}} & = & 4\pi M_{0,0,0,\mathbf{0}}(t,\boldsymbol{x}), \end{array} \tag{5.43}$$

where the moments of the NDF now read:

$$M_{i,j,k,\boldsymbol{l}}(t,\boldsymbol{x}) = \int_{\widehat{\overline{\boldsymbol{\xi}}^v} \in\Omega_\xi} H^i S^j v^k {V_{I,x}}^{l_x} {V_{I,y}}^{l_y} {V_{I,z}}^{l_z} f(t,\boldsymbol{x};\widehat{\overline{\boldsymbol{\xi}}^v})\, d^5\widehat{\overline{\boldsymbol{\xi}}^v}, \tag{5.44}$$

and $\boldsymbol{l} = (l_x, l_y, l_z)$. By considering the corresponding moments of the dynamics of the NDF (5.41), we can obtain the equations of the three expected interfacial variables and the volume fraction:

$$\begin{aligned}
\partial_t \alpha + \partial_{\boldsymbol{x}} \cdot \{\alpha(t,\boldsymbol{x})\widetilde{\mathbf{v}_I}\} &= \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (\{\alpha(t,\boldsymbol{x})(\widetilde{\mathbf{v}_I} - \widetilde{\mathbf{v}_I}^{\alpha})\}) + \int_{\widehat{\overline{\boldsymbol{\xi}}^{v}}} < \dot{v} >_c f(t,\boldsymbol{x};\widehat{\overline{\boldsymbol{\xi}}^{v}}) d^5\widehat{\overline{\boldsymbol{\xi}}^{v}}, \\
\partial_t \Sigma + \partial_{\boldsymbol{x}} \cdot \{\Sigma(t,\boldsymbol{x})\widetilde{\mathbf{v}_I}\} &= \int_{\widehat{\overline{\boldsymbol{\xi}}^{v}}} < -\dot{S} >_c f(t,\boldsymbol{x};\widehat{\overline{\boldsymbol{\xi}}^{v}}) d^5\widehat{\overline{\boldsymbol{\xi}}^{v}}, \\
\partial_t \Sigma\widetilde{H} + \partial_{\boldsymbol{x}} \cdot \{\Sigma\widetilde{H}\widetilde{\mathbf{v}_I}\} &= \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\{\Sigma\widetilde{H}(\widetilde{\mathbf{v}_I} - \widetilde{\mathbf{v}_I}^{H})\}\right) + \int_{\widehat{\overline{\boldsymbol{\xi}}^{v}}} < -\dot{S}\widehat{H} + \dot{H}\widehat{S} >_c f(t,\boldsymbol{x};\widehat{\overline{\boldsymbol{\xi}}^{v}}) d^5\widehat{\overline{\boldsymbol{\xi}}^{v}}, \\
\partial_t \Sigma\widetilde{G} + \partial_{\boldsymbol{x}} \cdot \{\Sigma\widetilde{G}\widetilde{\mathbf{v}_I}\} &= \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot \left(\{\Sigma\widetilde{G}(\widetilde{\mathbf{v}_I} - \widetilde{\mathbf{v}_I}^{G})\}\right).
\end{aligned} \tag{5.45}$$

The divergence terms appearing in the right-hand side of the system express the correlation between the velocity and the rest of the phase space variables. In the case where the velocity is uncorrelated with $\{v, H, G\}$, these terms vanish. Under this hypothesis, the only remaining source terms are related to the shape deformation and to the compressibility of the particles. Also, the last equation would have no source term: the deformation of particles does not affect the quantity $\Sigma\widetilde{G}$. Indeed, according to Gauss-Bonnet this quantity is the expected number density of particles, the evolution of which is only due to convection or coalescence and break-up of the particles. Since we have supposed that the particles do not break-up nor coalesce, the particles number density is simply convected at velocity $\widetilde{\mathbf{v}_I}$.

In the following section, we propose a simplified closed model of this system in the case of a polydisperse evaporating spray, where the droplets are supposed to remain spherical at any time.

5.4 High order geometrical size moments for polydisperse evaporating sprays of spherical droplets

Remark 1 *From now on, we omit the subscript* $\widehat{\bullet}$ *for the phase-space variables.*

In this part, we show that the system of equations given in (5.45) can degenerate to the fractional moments model described in chapter 4 in the case of spherical droplets and under the same assumptions as the ones used in that chapter. In the following, we suppose that the spray is dilute and the Weber number is low enough, so that coalescence and fragmentation of the droplets can be neglected. Finally, we assume that thermal transfer can also be neglected, so that the temperature of the droplets can be ignored. Under the hypothesis that the droplets stay spherical at all times, the NDF writes:

$$f(t,\boldsymbol{x};H,S,v,\mathbf{v}_I) = N(t,\boldsymbol{x};S,\mathbf{v}_I)\,\delta(H - \mathsf{H}(S))\,\delta(v - \mathsf{v}(S)), \tag{5.46}$$

where the mean curvature, $\mathsf{H}(S) = \sqrt{4\pi/S}$, and the volume of the drops, $\mathsf{v}(S) = S^{3/2}/(6\sqrt{\pi})$, are two functions of the surface S. Therefore, we can reduce the phase-space variables to the surface area S and the averaged interface velocity $\mathbf{v}_I$. Let us also note that under the assumption of the incompressibility of the droplet and a uniform evaporation at the droplet surface, $\mathbf{v}_I$ is now equal to $\boldsymbol{v}_p$, the velocity of the mass center of the droplet. In the following, we consider dimensionless variables, so that the surface area $S \in [0,1]$. The droplet acceleration $\dot{\boldsymbol{v}_p}$ is equal to the sum of forces per unit of mass acting on the droplet. In the following, we consider that the only force acting on the droplets is the drag due to the carrier gas. This can be modeled in a first time by a linear Stokes law:

$$\dot{\boldsymbol{v}}_p = \frac{\boldsymbol{u}_g - \boldsymbol{v}_p}{\mathrm{St}(S)}, \tag{5.47}$$

where $\mathrm{St}(S) = \theta S$ is the Stokes number, which depends linearly on the surface area S. The coefficient θ depends on the physical properties of the gas and the liquid. Since we consider spherical and non-deforming droplets, the evolution of the surface area is only due to evaporation. The Lagrangian-time derivative of the surface area is equal to the evaporation:

$$\dot{S} = \mathrm{R}_S(S). \tag{5.48}$$

Assuming a d^2 law for the phenomenon, the evaporation rate is constant: $\mathrm{R}_S(S) = -\mathrm{K}$. Considering the spherical assumption of the droplets, we can show the following relations:

$$\begin{aligned} \dot{v} &= -\frac{3}{2}\frac{1}{6\sqrt{\pi}}S^{1/2}\mathrm{K} \\ \dot{H} &= \frac{1}{2}\sqrt{4\pi}\frac{1}{S^{3/2}}\mathrm{K} \end{aligned} \tag{5.49}$$

To simplify the distribution on the velocity, we consider a monokinetic assumption de Chaisemartin et al.(2009):

$$N(t,\boldsymbol{x};S,\boldsymbol{v}_p) = n(t,\boldsymbol{x},S)\,\delta(\boldsymbol{v}_p - \boldsymbol{u}_p(t,\boldsymbol{x})), \tag{5.50}$$

The averaged interfacial quantities (volume fraction, surface area density and Gauss and mean curvatures), that have been expressed as moments of the NDF $f(t,\boldsymbol{x};H,S,v,\mathbf{v}_I)$ in equations (5.43) and (5.44), can now be expressed as moments of the simplified NDF $n(t,\boldsymbol{x},S)$ as follows:

$$\begin{aligned} \Sigma_d\widetilde{G}_d &= 4\pi\mathrm{m}_0, \\ \Sigma_d\widetilde{\mathrm{H}}_d &= 2\sqrt{\pi}\mathrm{m}_{1/2}, \\ \Sigma_d &= \mathrm{m}_1, \\ \alpha_d &= \frac{1}{6\sqrt{\pi}}\mathrm{m}_{3/2}, \end{aligned} \tag{5.51}$$

where a fractional size-moment is given by:

$$\mathrm{m}_{k/2}(t,\boldsymbol{x}) = \int_0^1 S^{k/2} n(t,\boldsymbol{x},S)dS. \tag{5.52}$$

Now, from equations (5.45), (5.51), (5.48), (5.49) and (5.47), we can derive the following system of equations on the fractional size-moments:

$$\begin{cases} \partial_t \mathrm{m}_0 & + \ \partial_{\boldsymbol{x}} \cdot \left(\mathrm{m}_0 \boldsymbol{u}_p\right) & = \ -\mathrm{K}\ (t,\boldsymbol{x},S=0), \\ \partial_t \mathrm{m}_{1/2} & + \ \partial_{\boldsymbol{x}} \cdot \left(\mathrm{m}_{1/2} \boldsymbol{u}_p\right) & = \ -\frac{\mathrm{K}}{2}\mathrm{m}_{-1/2}, \\ \partial_t \mathrm{m}_1 & + \ \partial_{\boldsymbol{x}} \cdot \left(\mathrm{m}_1 \boldsymbol{u}_p\right) & = \ -\mathrm{K}\mathrm{m}_0, \\ \partial_t \mathrm{m}_{3/2} & + \ \partial_{\boldsymbol{x}} \cdot \left(\mathrm{m}_{3/2} \boldsymbol{u}_p\right) & = \ -\frac{3\mathrm{K}}{2}\mathrm{m}_{1/2}, \\ \partial_t \left(\mathrm{m}_1 \boldsymbol{u}\right) & + \ \partial_{\boldsymbol{x}} \cdot \left(\mathrm{m}_1 \boldsymbol{u}_p \otimes \boldsymbol{u}_p\right) & = \ -\mathrm{K}\mathrm{m}_0 \boldsymbol{u}_p + \mathrm{m}_0 \frac{\boldsymbol{u}_g - \boldsymbol{u}_p}{\theta}, \end{cases} \tag{5.53}$$

where the pointwise disappearance flux $-\mathrm{K}n(t,\boldsymbol{x},S=0)$ at size zero and the negative order moment $\mathrm{m}_{-1/2} = \int_0^1 S^{-1/2} n(t,S)dS$ depend on the size distribution, which is an unknown function in this model.

Fractional size moments can be seen as a first and a simple model that we can derive successfully from the statical gas-liquid interface approach introduced in 5.3.1. In this model, we have supposed a spherical form of droplets which allows to simplify the system of equations (5.45), and thus obtain the closed system (5.53) using the reconstruction of the size distribution through entropy maximization. In this case, droplet deformation is neglected and the evaporation is the only source term acting on the size variation. So on, the actual model can be used for polydisperse evaporating sprays, but with more capacity to be coupled, in future work, with a model of the form (5.45) to describe the gas-liquid interface.

5.5 Unified averaged surface density function for disperse and separated phases

In section 5.3.1, we have drawn a link between the NDF and the discrete SDF in the context of a discrete formalism. However, the discrete SDF is valid only for the disperse phase and its definition (5.30) supposes that we are able to isolate the droplets/ligaments in a certain manner. Therefore, it is not yet obvious that we can link a statistical description of the gas-liquid interface with a statistical description of the spray evolving under Williams-Boltzmann's equation. In this part, we introduce a spatially averaged SDF, which is defined independently of the flow regime (disperse or separated phases). We show that the defined SDF degenerates to the discrete SDF (5.30), when the liquid phase is dispersed and dilute. The objective of this task is twofold: first, to draw a clear link between a distribution on the surface, which describes general gas-liquid interfaces, and a distribution of a droplet number density. The second purpose is to use this definition to design a new tool and new algorithms to analyze the gas-liquid interface generated by DNS computations.

5.5.1 Averaged interfacial quantities and appropriate phase space variables

The spatial averaging process is applied on the internal variables $\boldsymbol{\xi}(t,\boldsymbol{x})$ and the instantaneous area concentration $\delta_I(t,\boldsymbol{x})$ separately for each realization. In this context, we define generally the Averaged SDF (ASDF) as follows:

$$\mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi}) =< \widetilde{\delta_I}(t,\boldsymbol{x})\,\delta(\boldsymbol{\xi}-\overline{\boldsymbol{\xi}}(t,\boldsymbol{x})) >, \tag{5.54}$$

where the averaged space variables $\overline{\boldsymbol{\xi}}(t,\boldsymbol{x})$ and the average area measure $\widetilde{\delta_I}(t,\boldsymbol{x})$ will be defined such that we satisfy the following properties:

A. The spatial average of the surface area does not spread the interface: its thickness remains zero for each realization, *i.e.*

$$\widetilde{\delta_I}(t,\boldsymbol{x}) = 0, \qquad \text{for } \boldsymbol{x} \notin \Sigma_I(t).$$

B. The new distribution preserves the space integral of the first order moments of the classical SDF as in Proposition 5.3.1. This property reads:

$$\int_{\boldsymbol{x}\in\Omega_x}\int_{\boldsymbol{\xi}\in\Omega_\xi} \boldsymbol{\xi}^l \mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi})d^5\boldsymbol{\xi}d^3\boldsymbol{x} = \int_{\boldsymbol{x}\in\Omega_x}\int_{\boldsymbol{\xi}\in\Omega_\xi} \boldsymbol{\xi}^l F(t,\boldsymbol{x};\boldsymbol{\xi})d^5\boldsymbol{\xi}d^3\boldsymbol{x} \tag{5.55}$$

where $\boldsymbol{l} \in \{0,1\}^5,\ l_1+\cdots+l_5 \leq 1$ and $\boldsymbol{\xi} = \{\xi_1, \dots, \xi_5\}$. We recall that this property ensures the possibility to express the expected-mean interfacial quantities $(\Sigma(t,\boldsymbol{x}), \widetilde{H}(t,\boldsymbol{x}), \widetilde{G}(t,\boldsymbol{x}))$ as moments of the new averaged SDF.

C. The new SDF can be related to the discrete SDF, when the domain contains a dilute disperse phase, such that the larger particle diameter is smaller than the inter-particle distance. We express this relation for a volume space $E \subset \Omega_x$, where the border of the domain does not cross any particle, as follows:

$$\int_{\boldsymbol{x}\in E} \mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi}) d^3\boldsymbol{x} = \int_{\boldsymbol{x}\in E} F^d(t,\boldsymbol{x};\boldsymbol{\xi}) d^3\boldsymbol{x}. \tag{5.56}$$

In these conditions, we can relate the spatial-averaged SDF to the NDF of the particles in the same way as in section 5.3.2.

To fulfill the first requirement A, we propose to define the averaged quantities as follows:

$$\widetilde{\delta_I}(\boldsymbol{x}) = \Theta_h(t,\boldsymbol{x})\delta_I(t,\boldsymbol{x}), \tag{5.57}$$

$$\overline{\boldsymbol{\xi}}(t,\boldsymbol{x}) = \begin{cases} \dfrac{1}{\Theta_h(t,\boldsymbol{x})} \int_{\boldsymbol{y}\in\Omega_x} w(\boldsymbol{x};\boldsymbol{y}-\boldsymbol{x})\, \delta_I(t,\boldsymbol{y})\boldsymbol{\xi}(t,\boldsymbol{y}) d^3\boldsymbol{y}, & \text{if } \Theta_h(t,\boldsymbol{x}) > 0 \\ \\ \text{Not defined} & \text{otherwise} \end{cases} \tag{5.58}$$

and

$$\Theta_h(t,\boldsymbol{x}) = \int_{\boldsymbol{y}\in\Omega_x} w(\boldsymbol{x};\boldsymbol{y}-\boldsymbol{x})\, \delta_I(t,\boldsymbol{y}) d^3\boldsymbol{y}, \tag{5.59}$$

where $w(\boldsymbol{x};\boldsymbol{r} = \boldsymbol{y}-\boldsymbol{x})$ is a convolution kernel. The function $w(\boldsymbol{x};\boldsymbol{r})$ is chosen such that it vanishes for large $||\boldsymbol{r}|| > h$, where $h > 0$ is a characteristic spatial length scale, independent from the position $\boldsymbol{x}$. The case where $\Theta_h(t,\boldsymbol{x}) = 0$, the averaged interfacial variables are not defined and can take any real value, since no interface exists in the vicinity of the position $\boldsymbol{x}$.

The kernel function is designed such that the SDF (5.54), satisfies the two remaining requirements (B and C).

For the second requirement B, we can show that equation (5.55) is satisfied if we have, for any $\boldsymbol{y} \in \Omega_x$:

$$\int_{\boldsymbol{x}\in\Omega_x} w(\boldsymbol{x},\boldsymbol{y}-\boldsymbol{x})\delta_I(t,\boldsymbol{x}) d^3\boldsymbol{x} = 1. \tag{5.60}$$

Finally, for the third requirement property C, we look for a spatial-average of the interfacial variables (5.58) that can degenerate to an interfacial average over one particle surface (5.29), when the inter-particle distance is much larger than the particles diameters. We can reach this goal by first choosing a length scale h larger than the maximum of particle diameters and smaller than the inter-particle distance. Then we propose to use the following kernel function, which satisfies the condition (5.60):

$$w(\boldsymbol{x};\boldsymbol{r}) = \frac{\mathbb{1}_{||\boldsymbol{r}||<h}(\boldsymbol{r})}{\int_{\boldsymbol{x}'\in\mathcal{V}_h(\boldsymbol{x}+\boldsymbol{r})} \delta_I(\boldsymbol{x}') d^3\boldsymbol{x}'}, \tag{5.61}$$

where $\mathcal{V}_h(\boldsymbol{x}) = \{\boldsymbol{y} \in \Omega_x;\ ||\boldsymbol{x}-\boldsymbol{y}|| < h\}$. Therefore, $w(\boldsymbol{x},\boldsymbol{r})$ represents the inverse of the surface area included in $\mathcal{V}_h(\boldsymbol{x}+\boldsymbol{r})$. In the case where no interface exists in the volume $\mathcal{V}_h(\boldsymbol{x}+\boldsymbol{r})$, $w(\boldsymbol{x},\boldsymbol{r})$ is not defined and can take any real value without affecting the definition of the averaged SDF.

5.5.2 Generalized Number density function

Using the averaged SDF defined in the previous section, we can define the following distribution function:

Definition We define the Generalized Number Density Function (GNDF) as follows:

$$f_\Sigma(t,\boldsymbol{x};\boldsymbol{\xi}) = \frac{|G|}{4\pi}\mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi}). \tag{5.62}$$

Now, let us consider a disperse phase made of $N_{\max}$ non-spherical particles (droplets or bubbles), $p_k,\ k = 1,\dots,N_{\max}$ whose external boundaries are denoted by $\Sigma_k,\ k = 1,\dots,N_{\max}$ and their center of mass by $\boldsymbol{x}_k,\ k = 1,\dots,N_{\max}$. By considering a sufficiently large length scale averaging h (for example larger than the particle diameter), we can ensure the positivity of the averaged Gauss curvature. In this case, we can simply write the GNDF as follows:

$$f_\Sigma(t,\boldsymbol{x};\boldsymbol{\xi}) = \frac{G}{4\pi}\mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi}). \tag{5.63}$$

In the following, we consider a dilute disperse phase, such that the inter-distance between the droplets is larger than the droplet diameter. Using a length scale h larger than the droplet diameter, we show the link between the GNDF and the classic NDF, as it was stated in the third requirement C of the last section. To conduct this derivation, we use the notations of section 5.3.1, which introduced the discrete formalism of the particles.

Proposition 5.5.1 *If E is a space volume whose borders do not cross any particle, then the two integrals of the GNDF $f_\Sigma(t,\boldsymbol{x};\boldsymbol{\xi}) = \frac{G}{4\pi}\mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi})$ and the NDF $f(t,\boldsymbol{x};\boldsymbol{\xi})$ ever E are equal:*

$$\int\limits_{\boldsymbol{x}\in E} f_\Sigma(t,\boldsymbol{x};\boldsymbol{\xi})d^3\boldsymbol{x} = \int_{\boldsymbol{x}\in E} f(t,\boldsymbol{x};\boldsymbol{\xi})d^3\boldsymbol{x}. \tag{5.64}$$

Proof: If h is larger than the largest particle diameter and smaller than the smallest inter-particle distance, then $\omega_k = \{\boldsymbol{x} \in \Omega_x,\ \min\limits_{\boldsymbol{y}\in\Sigma_k} ||\boldsymbol{x}-\boldsymbol{y}|| \le h\}$ is a subdomain of Ω_x which contains only the particle p_k. Also, all the interfacial information is contained within the intersection of these subdomains $\cap_{k=1}^{N_{\max}}\omega_k$, so that one can write:

$$\begin{aligned}
\int\limits_{\boldsymbol{x}\in E} G\mathrm{F}^a(t,\boldsymbol{x};\boldsymbol{\xi})d^3\boldsymbol{x} &= <\sum_{k=1}^{N_{\max}}\left\{\int\limits_{\omega_k\cap E} G\,\widetilde{\delta_I}(t,\boldsymbol{x})\,\delta(\boldsymbol{\xi}-\overline{\boldsymbol{\xi}}(t,\boldsymbol{x})d^3\boldsymbol{x}\right\}>,\\
&= <\sum_{k=1}^{N_{\max}}\left\{\int\limits_{\omega_k\cap E} \overline{G}_k\,\widetilde{\delta_I}(t,\boldsymbol{x})\,\delta(\boldsymbol{\xi}-\overline{\boldsymbol{\xi}}_k)d^3\boldsymbol{x}\right\}>,\\
&= <\sum_{\{k,\boldsymbol{x}_k\in E\}}\left\{\overline{G}_k\,S_k\,\delta(\boldsymbol{\xi}-\overline{\boldsymbol{\xi}}_k)\right\}>,\\
&= 4\pi<\sum_{\{k,\boldsymbol{x}_k\in E\}}\left\{\delta(\boldsymbol{\xi}-\overline{\boldsymbol{\xi}}_k)\right\}>,\\
&= 4\pi\int_{\boldsymbol{x}\in E} f(t,\boldsymbol{x};\boldsymbol{\xi})d^3\boldsymbol{x},
\end{aligned} \tag{5.65}$$

To pass from the third to the fourth equality, we have used Gauss-Bonnet formula and have supposed that all droplets are homeomorphic to a sphere. □

This result shows a general interpretation for the NDF, which is not restricted to a discrete formalism. In the same way as in section 5.3.1, we can derive the GPBE satisfied by the GNDF:

$$\partial_t f_\Sigma + \nabla_{\boldsymbol{x}} \cdot \{\mathbf{v}_I f_\Sigma\} + \nabla_\xi \cdot \{< \dot{\xi} >_c f_\Sigma\} =< \Gamma >_c f_\Sigma, \tag{5.66}$$

where $\Gamma = \dot{\mathcal{S}} + \frac{\dot{G}}{G}$ is a source term of topology variation and which can be related mainly to the breakup and coalescence phenomena.

Now, it is more obvious that the SDF and NDF are strongly related. The SDF can be seen as the origin of the NDF as it was discussed in Canu et al.(2017). The averaged SDF allows to consider a generic statistical description of the gas-liquid interface, which can also be directly related to the NDF in the case of dilute disperse phase, as it is stated by proposition 5.5.1. Indeed, by using the spatial averaging procedure, we are able to isolate droplets in dilute disperse phase region, and thus we can compute the NDF from the averaged SDF.

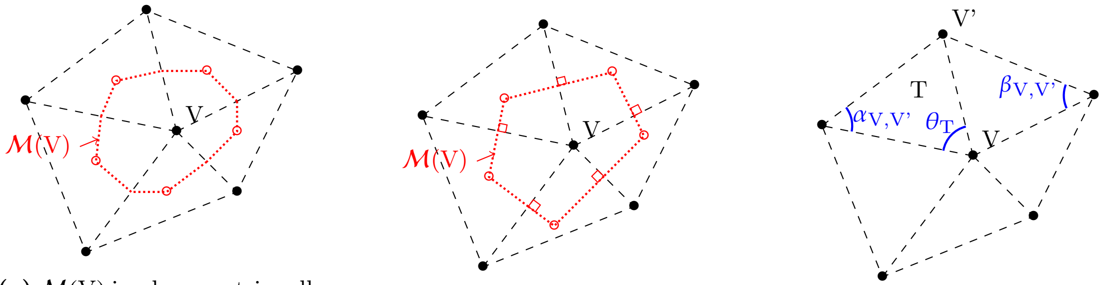
5.6 Algorithms and techniques for the numerical computation of the curvatures and of the SDF

In this part, we apply the results of the previous section to the design of a new numerical tool dedicated to the post-processing of DNS two-phase flows simulations. Such tool allows to enhance the analysis of the gas-liquid interface evolution. The DNS computations are performed using the ARCHER code Menard et al.(2007), where a combined VOF and level-set approach is used to capture the interface, and a ghost method is applied to represent accurately the jump of variables across the liquid-gas interface.

Some numerical tools to compute the curvatures and the surface area of the gas-liquid interface present in each cell are already available in the ARCHER code Canu et al.(2017). Unfortunately, these quantities being computed separately, the Gauss-Bonnet formula can not be numerically satisfied. For this reason, we choose to compute the curvatures and the surface element areas by using the algorithm presented thereafter. Such algorithm will allow to obtain the number density function of droplets in DNS simulations, by using simple computations on the surface and without the need of an algorithm that isolates the droplets Kang et al.(2000).

The new algorithm is implemented independently from the ARCHER code, from which only the distance function (Level-Set) data is used. The different algorithm steps are summarized as follows:

- The gas-liquid interface is discretized with a 2D triangulated mesh using the Marching Cubes algorithm Thomas et al.(2003). This algorithm takes the $3D$ level-set scalar field as an input and returns a 2D meshed surface. This mesh is described with two arrays: the array of vertices, $\mathbf{V}$ of dimension $n_v \times 3$, and the array of faces (defined by three vertices), $\mathbf{F}$ of dimension $n_f \times 3$ which defines the connectivity between vertices. In this work, we use the Python package sckimage Van der Walt et al.(2014) to triangulate the surface.
- For each vertex $\mathrm{V} \in [1, n_v]$, $\mathbf{O}(\mathrm{V})$ denotes the set of neighbors of V: V′ is a neighbor of V if both vertices share a same face $\mathrm{f} \in [1, n_f]$. By abuse of notation, $\mathbf{O}(\mathrm{V})$ also denotes the set of faces V belongs to and for each $\mathrm{f} \in \mathbf{O}(\mathrm{V})$, $\boldsymbol{n}(\mathrm{f})$ denotes the normal vector to the face, oriented toward the gas phase.
- In the neighbourhood of V, we define its dual cell $\mathcal{M}(\mathrm{V})$ as a mix between the Voronoi cell (Figure 5.2b) and the barycentric cell (Figure 5.2a) around V, as proposed in Meyer



(a) $\mathcal{M}$(V) is a barycentric cell. **(b)** $\mathcal{M}$(V) is a Voronoi cell. **(c)** Illustration of angles.

Figure 5.2: *Neighboring vertices and surface elements $\mathcal{M}(V)$ around the vertex V.*

et al.(2001). Simply, for each triangular face f containing V, the Voronoi contour is considered if all the angle are acute. The barycentric contour is considered otherwise. Also, θ_{f} denotes the angle at vertex V in f, see Figure 5.2c, and $A[\text{f}, \text{V}] = \text{area}(\text{f} \cap \mathcal{M}(\text{V}))$ is the area of the dual cell within the face f. Then, the discrete local area $A[\text{V}]$, normal $\boldsymbol{n}[\text{V}]$ and Gauss curvature $G[\text{V}]$ can be computed:

$$A[\text{V}] = \sum_{\text{f} \in \mathbf{O}(\text{V})} A[\text{f}, \text{V}], \qquad \boldsymbol{n}[\text{V}] = \frac{\sum_{\text{f} \in \mathbf{O}(\text{V})} A[\text{f}, \text{V}]\boldsymbol{n}(\text{f})}{\left\| \sum_{\text{f} \in \mathbf{O}(\text{V})} A[\text{f}, \text{V}]\boldsymbol{n}(\text{f}) \right\|}, \tag{5.67}$$

$$G[\text{V}] = \left(2\pi - \sum_{\text{f} \in \mathbf{O}(\text{V})} \theta_{\text{f}} \right) / A[\text{V}].$$

- Finally, for each neighbour V′ of V, $\alpha_{\text{V,V}'}$ and $\beta_{\text{V,V}'}$ denote the two angles facing the edge $[\text{V}, \text{V}']$, without distinction, as illustrated in Figure 5.2c. Now, the local discrete mean curvature reads:

$$H[\text{V}] = \frac{1}{2A[\text{V}]} \sum_{\text{V}' \in \mathbf{O}(\text{V})} \Big(\cot(\alpha_{\text{V}',\text{V}}) + \cot(\beta_{\text{V}',\text{V}}) \Big) \left(\mathbf{V}[\text{V}] - \mathbf{V}[\text{V}'] \right) \cdot \boldsymbol{n}[\text{V}]. \tag{5.68}$$

After computing the local geometrical quantities, we can evaluate the fine-grain localized SDF for one simulation, *i. e.* one realization. In the following, the phase space variables are restricted to the two curvatures $\boldsymbol{\xi} = (H, G) \in \Omega_\xi$: we do not consider the interface velocity. This 2D space is discretized into $n_h \times n_g$ phase-space cells $C_{i,j}$, such that the discrete phase space variables are expressed as follows:

$$\boldsymbol{\xi}_{i,j} = (H_i, G_j) = \boldsymbol{\xi}_{min} + (i\Delta H, j\Delta G), \quad (i,j) \in [1, n_h] \times [1, n_g],$$

where $\Delta H = (H_{max} - H_{min})/n_h$, $\Delta G = (G_{max} - G_{min})/n_g$ and the subscripts min and max refer respectively to the minimum and maximum values obtained by (5.67) and (5.68). The integral of the fine-grain localized SDF over a subdomain $E \subset \Omega_x$ can be approximated by:

$$\int_{\boldsymbol{x} \in E} F'(t, \boldsymbol{x}; \boldsymbol{\xi}_{i,j}) d^3\boldsymbol{x} \simeq \frac{1}{\Delta H \Delta G} \sum_{\text{V}=1}^{n_v} A[\text{V}]\, I_{C_{i,j}}(\boldsymbol{\xi}[\text{V}])\, I_E(\mathbf{V}[\text{V}]) \tag{5.69}$$

where $F'(t, \boldsymbol{x}; \boldsymbol{\xi}_{i,j})$ is the fine-grain SDF defined in (5.18), $I_{C_{i,j}}(.)$ is the characteristic function of the cell $C_{i,j}$:

$$C_{i,j} = \{\boldsymbol{\xi}', |H' - H_{i,j}| \leq \Delta H/2 \text{ and } |G' - G_{i,j}| \leq \Delta G/2\},$$

and $I_E(.)$ is the characteristic function of the sub-domain E. In the following, we take $E = \Omega_x$. Thus the last equation becomes:

$$\int_{\boldsymbol{x}\in\Omega_x} F^{'}(t,\boldsymbol{x};\xi_{i,j})d^3\boldsymbol{x} \simeq \frac{1}{\Delta H \Delta G}\sum_{\mathrm{V}=1}^{n_v} A[\mathrm{V}]\, I_{C_{i,j}}(\boldsymbol{\xi}[\mathrm{V}]). \tag{5.70}$$

Using this numerical approximation, we define the numerical fine-grain SDF integrated in the whole domain as follows:

$$\mathcal{F}^{'}(\xi_{i,j}) = \frac{1}{\Delta H \Delta G}\sum_{\mathrm{V}=1}^{n_v} A[\mathrm{V}]\, I_{C_{i,j}}(\boldsymbol{\xi}[\mathrm{V}]). \tag{5.71}$$

To evaluate the averaged fine-grain SDF (5.54), we first calculate the averaged area and curvatures at each vertex $\mathrm{V} \in [1, n_v]$. These averaged quantities depend on the length scale h. In the following, we take $h = (k/2)\Delta x$, where Δx is the size of the level-set computation cells and $k \geq 1$ is an integer value to be set by the user. We denote by $\widetilde{A}^k[\mathrm{V}]$ and $\overline{\boldsymbol{\xi}}^k[\mathrm{V}]$ the numerical approximation of the averaged area and curvatures, defined in (5.57), (5.58) and (5.61) and which are numerically evaluated as follows:

$$\begin{array}{rcl} \widetilde{A}^k[\mathrm{V}] & = & \displaystyle\sum_{\mathrm{V}'\in\mathbf{O}_k(\mathrm{V})} \frac{A[\mathrm{V}']}{S_k[\mathrm{V}']}A[\mathrm{V}], \\ \overline{\boldsymbol{\xi}}^k[\mathrm{V}] & = & \displaystyle\frac{1}{\widetilde{A}^k[\mathrm{V}]}\sum_{\mathrm{V}'\in\mathbf{O}_k(\mathrm{V})} \boldsymbol{\xi}[\mathrm{V}']\frac{A[\mathrm{V}']}{S_k[\mathrm{V}']}A[\mathrm{V}], \end{array} \tag{5.72}$$

where $\mathbf{O}_k(\mathrm{V})$ is the set of vertices V', such that $||\mathbf{V}[\mathrm{V}] - \mathbf{V}[\mathrm{V}']||_2 \leq k/2\Delta x$ and

$$S_k[\mathrm{V}'] = \sum_{\mathrm{V}''\in O_k(\mathrm{V}')} A(\mathrm{V}'').$$

The numerical fine-grain averaged SDF is given by:

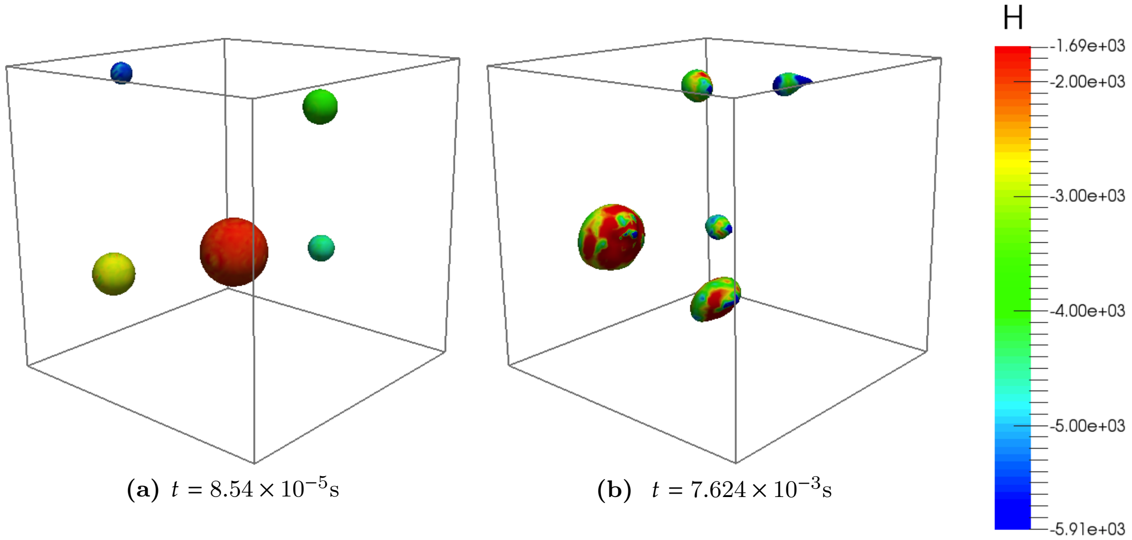
$$\mathcal{F}^{'a}(\xi_{i,j}) = \frac{1}{\Delta H \Delta G}\sum_{\mathrm{V}=1}^{n_v} \widetilde{A}^k[\mathrm{V}] I_{C_{i,j}}(\overline{\boldsymbol{\xi}}^k[\mathrm{V}]). \tag{5.73}$$

Then, the GNDF can be evaluated numerically as follows:

$$\mathcal{N}^{'}(\xi_{i,j}) = \frac{1}{\Delta H \Delta G}\sum_{\mathrm{V}=1}^{n_v} \frac{|\overline{G}^k[\mathrm{V}]|}{4\pi}\widetilde{A}^k[\mathrm{V}] I_{C_{i,j}}(\overline{\boldsymbol{\xi}}^k[\mathrm{V}]). \tag{5.74}$$

We recall that the spatial averaging of the curvatures and surface area distribution introduced in section 5.5.1, has been designed such that the first order moments of the SDF (5.55) are preserved. We can verify this property numerically as follows:

$$\begin{array}{rcccl} \int_{\Omega_x}\Sigma(t,\boldsymbol{x})d\boldsymbol{x} & \simeq & \sum_{\mathrm{V}} A[\mathrm{V}] & = & \sum_{\mathrm{V}} \widetilde{A}^k[\mathrm{V}], \\ \int_{\Omega_x}\Sigma\widetilde{H}(t,\boldsymbol{x})d\boldsymbol{x} & \simeq & \sum_{\mathrm{V}} H[\mathrm{V}]A[\mathrm{V}] & = & \sum_{\mathrm{V}} \overline{H}^k[\mathrm{V}]\widetilde{A}^k[\mathrm{V}], \\ \int_{\Omega_x}\Sigma\widetilde{G}(t,\boldsymbol{x})d\boldsymbol{x} & \simeq & \sum_{\mathrm{V}} G[\mathrm{V}]A[\mathrm{V}] & = & \sum_{\mathrm{V}} \overline{G}^k[\mathrm{V}]\widetilde{A}^k[\mathrm{V}]. \end{array} \tag{5.75}$$



(a) $t = 8.54 \times 10^{-5}$s **(b)** $t = 7.624 \times 10^{-3}$s

Figure 5.3: *Droplet surface colored according to the mean curvature values at the surfaces for two different times.*

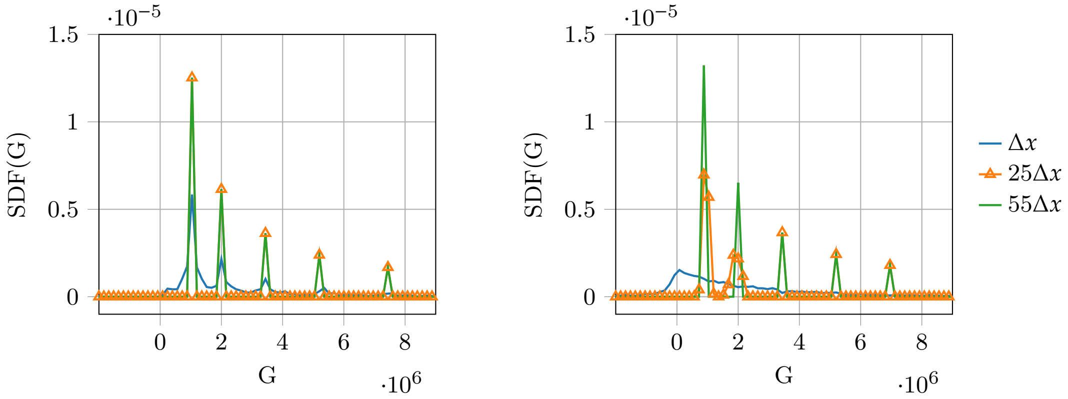
5.7 Numerical test with DNS simulations

In this part, we use the ARCHER code to perform some two-phase flows direct numerical simulations. The algorithms described in the previous section will be used to post-process the level-set data in order to compute the fine-grain SDF and NDF of a single simulation (one realization). Two numerical tests, with and without topological changes, allow to test the algorithm and the capacity to find some geometrical and topological feature of the gas-liquid interface, depending on a chosen averaging characteristic length-scale h defined in the previous section.

5.7.1 Droplets homeomorphic to spheres

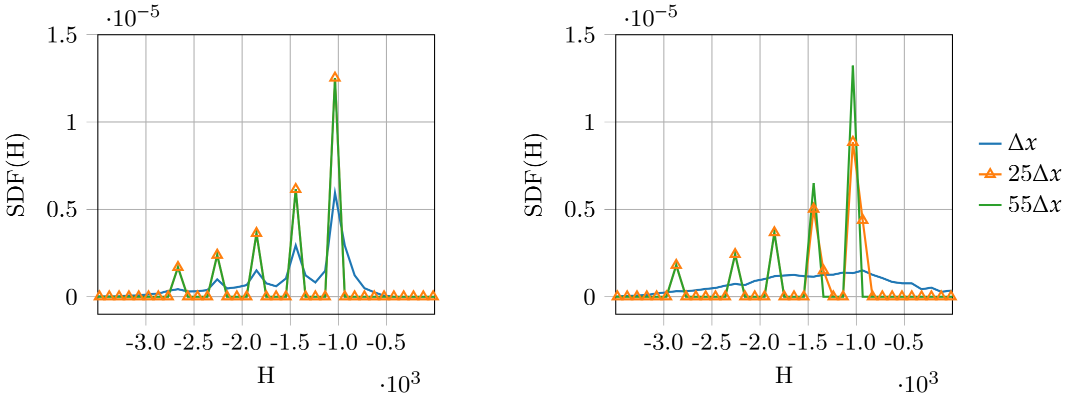
In the first configuration, we consider five initial spherical water droplets of radii $r \in [0.33 \text{ mm}, 1 \text{ mm}]$ injected at a velocity 5 m/s in a 1 cm^3 periodic cubic domain of initially still gas. Classical water/air properties are used here. The initial positions of the droplets centers are chosen randomly, with the constraint however that the inter-droplet distance d_{drop} (computed from the droplet surface) is larger than $2r_{max}$, where initially $r_{max} = 1$ mm. Due to the difference between the gas and the droplet velocities, the shape of the droplets is deformed in time without break-up (the maximum Weber number is $We_{max} < 1$). In Figure 5.3, we display an illustration of the droplets at two successive times. The color indication at the droplet surfaces shows an estimate of the mean curvature. This first case is envisioned in the framework of no topological changes.

In Figure 5.4 (respectively 5.5), we plot the SDF for different averaged scales as a function of the Gauss (respectively the mean) curvature. The localized SDF, plotted with a blue line, corresponds to the length scale average $h = \Delta x/2$. In this case, the SDF is continuous and we can not identify the surface area contribution of each droplet. But, by using larger scales of average ($h = 25\Delta x/2$ and $h = 55\Delta x/2$), we see that we can identify five peaks which correspond to the five droplets. Indeed, when $d_{drop} > h = 55\Delta x/2 > r_{max}$, for any point $\boldsymbol{x}$ situated at the interface of a droplet, the space volume $\mathcal{V}_h(\boldsymbol{x}) = \{\boldsymbol{y} \in \Omega_x;\ ||\boldsymbol{x} - \boldsymbol{y}|| < h\}$ contains that, and



(a) $t = 8.54 \times 10^{-5}$s **(b)** $t = 7.624 \times 10^{-3}$s

Figure 5.4: *Numerical SDF over the domain space as a function of the Gauss curvature: localized SDF (dashed-line), averaged SDF with $k = 25$ (triangle) and $k = 55$ (solid line).*

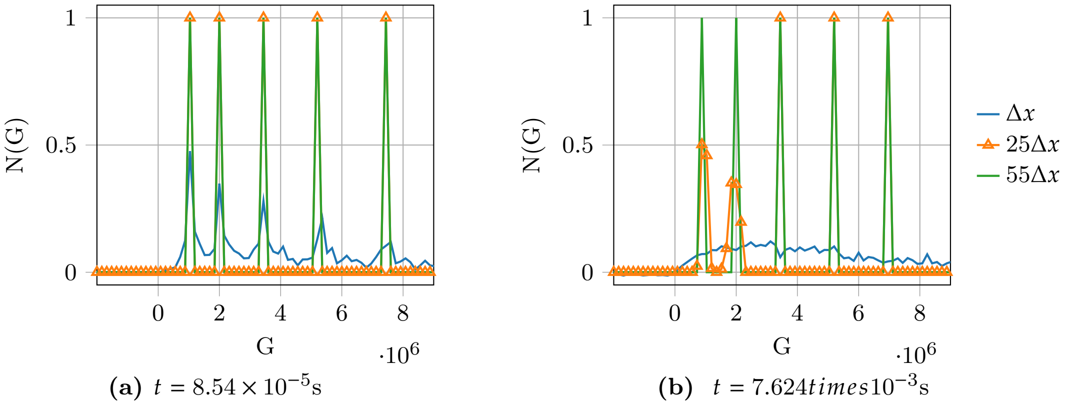


(a) $t = 8.54 \times 10^{-5}$s **(b)** $t = 7.624 \times 10^{-3}$s

Figure 5.5: *Numerical SDF over the domain space as a function of the mean curvature: localized SDF (dashed-line), averaged SDF with $k = 25$ (triangle) and $k = 55$ (solid line).*

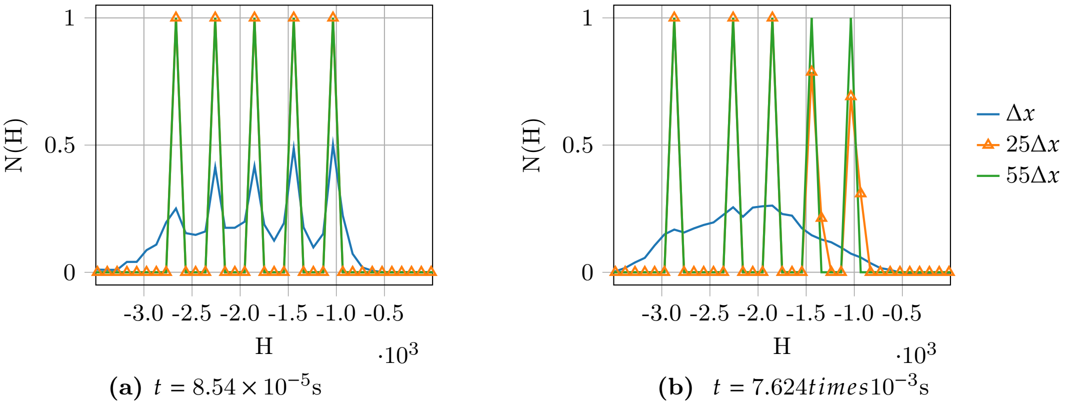
only that, droplet. Thus, for each droplet we compute one average mean and Gauss curvatures. Consequently, in this case the averaged SDF becomes a sum of Dirac delta function.

In Figure 5.6 (respectively 5.7), we display the GNDF for different averaging scales. These distributions are obtained from the averaged SDFs using (5.74). When the averaging scale h is larger than the droplet diameters, we obtain five peaks of value 1. Then, we can count the right number of droplets for each averaged Gauss or mean curvature thanks to Gauss-Bonnet formula with a proper evaluation of the curvature and using our averaging approach. This assesses the proposed theoretical and algorithmic approach within the framework of no topological change.



(a) $t = 8.54 \times 10^{-5}$s **(b)** $t = 7.624 times 10^{-3}$s

Figure 5.6: *Numerical GNDF over the domain space as a function of the Gauss curvature: localized GNDF (dashed-line), averaged GNDF with $k = 25$ (triangle) and $k = 55$ (solid line).*



(a) $t = 8.54 \times 10^{-5}$s **(b)** $t = 7.624 times 10^{-3}$s

Figure 5.7: *Numerical GNDF over the domain space as a function of the mean curvature: localized GNDF (dashed-line), averaged GNDF with $k = 25$ (triangle) and $k = 55$ (solid line).*

5.7.2 Two droplets collision

In this section, we consider a simulation of a two water droplets collision in the stretching separation regime Ashgriz and Poo(1990). The two droplets are initially separated (Figure 5.8a) in a periodic domain, with initial velocities. The collision leads to the temporary coalescence of the two droplets (Figure 5.8b) by forming a one unstable big droplet. Due to competition between inertial effect and surface tension, the unstable droplet deforms, first into a torus shape (Figure 5.8c) and finally breaks up into small different droplets (Figure 5.8d).
The parameters of collision yield some important topological changes and are given in the following Table 5.1:

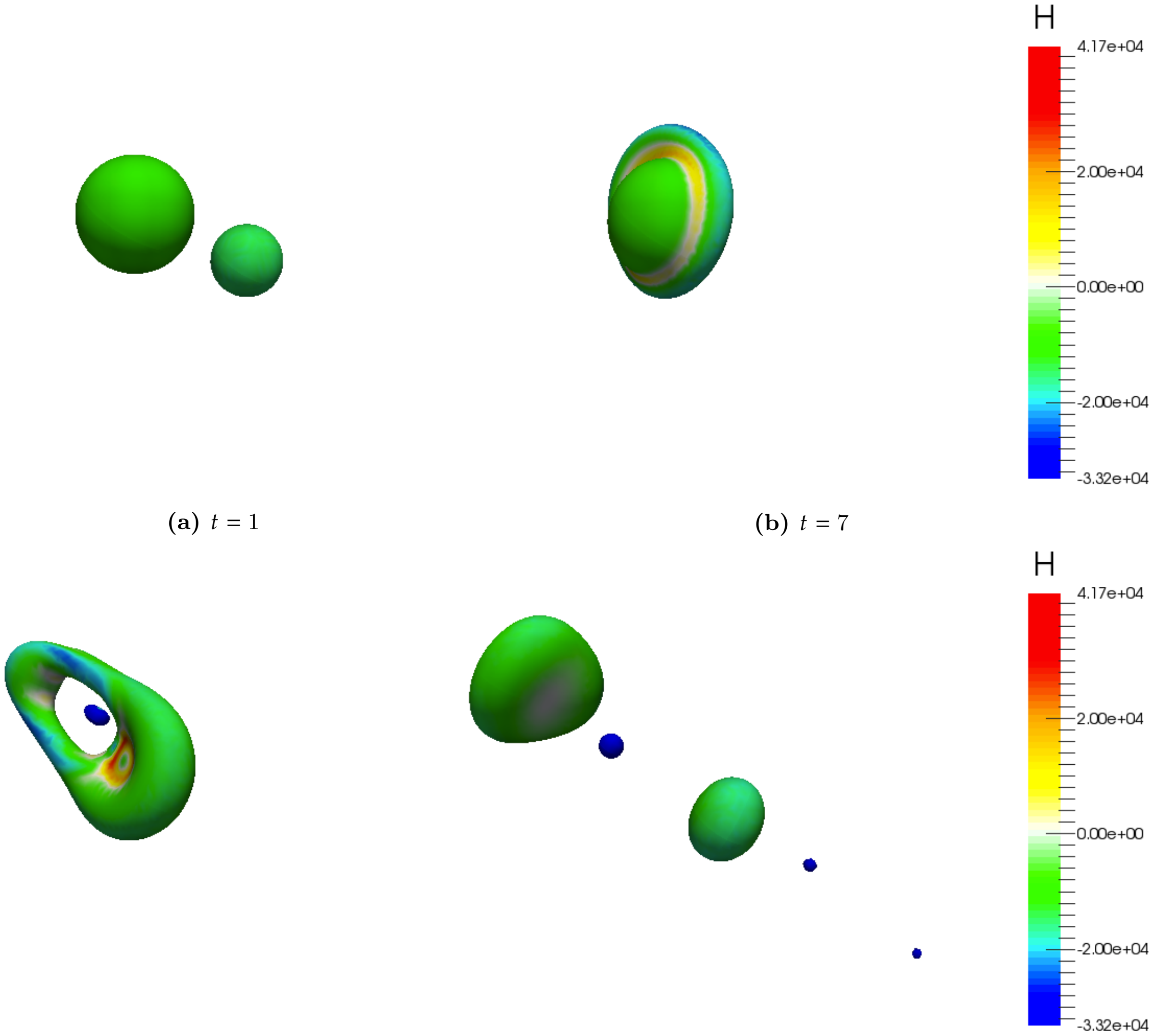
D_s (μm)	D_l (μm)	U_c $(m.s^{-1})$	$We = \frac{\rho_l U_c D_s}{\sigma}$	x
260	400	4	60	0.42

Table 5.1: *Parameters of droplet collision. Subscript s and l are related respectively to the small and large droplets. U_c is their relative velocity and x the dimensionless impact parameter Ashgriz and Poo(1990). Classical water and air properties are used.*

In this simulation, we compute the time evolution of the volume integral over the whole computational domain of the zero and first order moments of the localized and averaged SDFs (Σ, $\Sigma\widetilde{H}$, $\Sigma\widetilde{G}$). We can see from Figures 5.9-5.11, that the moments of the averaged and localized SDF are equal at all time. This equality corresponds to the requirement B. The first two Figures 5.9-5.10 correspond respectively to the total surface area and the total mean curvature. These two quantities evolve continuously and we can identify from the two curves the different droplet states:

- $t \in [0, 2]$: the two droplets are initially separated,
- $t \in [2, 12]$: coalescence and stretching of the two droplets are characterized by a minimum total surface area at $t \in [3, 4]$, just after the coalescence. Then a maximum surface area is reached at $t \in [11, 12]$, just before the breakup occurs, when the thin liquid film reaches its maximum length. In the same time, we obtain a minimum of the absolute value of the total mean curvature. This can be explained by the positive values of mean curvature during coalescence.
- $t \in [12, 40]$: breakup cascade and coalescence of some small droplets take place. The total surface area decreases while the absolute value of the mean curvature increases,
- $t \in [40, 80]$: in the final state, we obtain five droplets, where we have a convergence of the total surface area and the total mean curvature toward stable values.

The evolution of total Gauss curvature illustrated in Figure 5.11 is discontinuous and by dividing this quantity by 4π, we obtain integer values. Indeed, the quantity $1/(4\pi)\int_{\boldsymbol{x}} \Sigma\widetilde{G} d\boldsymbol{x}$ is equal to the sum of the half of the Euler characteristic of the objects included in the entire domain. This quantity will allow us to evaluate the topology evolution. In the case of droplets homeomorphic to spheres, we get the droplets number in the domain. This is the case here with the exception of the period of time between time $t = 10$ and $t = 20$, where this quantity drops down from 1 to -2 and then increases to 0 and 1 before reaching 2 again, with two objects homeomorphic to a sphere. During this interval of time, the big droplet formed by the coalescence deforms into a torus shape with several holes and with satellite droplets and then we come back to a regular torus with another droplet homeomorphic to a sphere, that is a total characteristic of 1, before the torus closes, that is a characteristic of 2, and then breaks into a total of three droplets at



(a) $t = 1$ **(b)** $t = 7$

(c) $t = 14$ **(d)** $t = 60$

Figure 5.8: *Simulation of collision and stretching separation of two droplets. Surface colored according to the local mean curvature (H) value.*

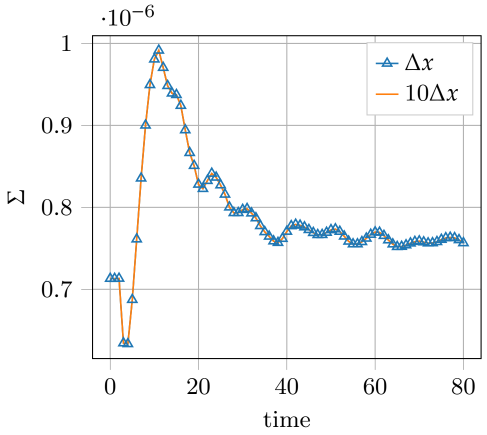


Figure 5.9: *Time evolution of the total surface area* $\int_{\boldsymbol{x}} \Sigma(\boldsymbol{x})d\boldsymbol{x}$*: without averaging (dashed-line), with scale average* $k = 20$ *(solid line).*

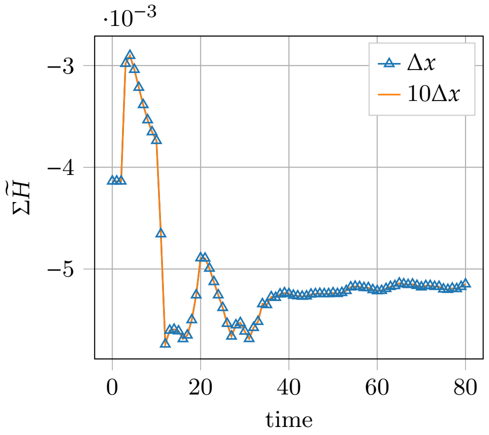


Figure 5.10: *Time evolution of the total mean curvature* $\int_{\boldsymbol{x}} \Sigma\widetilde{H}(\boldsymbol{x})d\boldsymbol{x}$*: without averaging (dashed-line), with scale average* $k = 20$ *(solid line).*

time 20.

Thus, the proposed approach not only can lead to a statistics of objects through spatial averaging, when the whole set of objects are homeomorphic to spheres, but also provides some key informations about the topology of the interface.

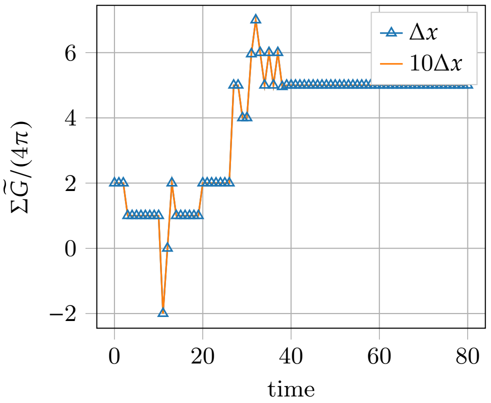


Figure 5.11: *Time evolution of the total gauss curvature* $\int_{x}\Sigma\widetilde{G}(\boldsymbol{x})d\boldsymbol{x}$*: without averaging (dashed-line), with scale average* $k = 20$ *(solid line).*

5.8 Conclusion

In this chapter, we have proposed a new statistical approach of the gas-liquid interface dedicated to two-phase flow modeling based on geometrical interface variables.

Relying on a statistical description of the interface between the two phases, the first contribution has been to propose a transport equation for a surface density function valid for both regimes: disperse and separated phases. The related phase space has been identified: it includes the curvatures and the velocity of the interface. An original link between such a surface density function formalism and its application to obtain statistics at the level of objects, such as the number density function (NDF) for sprays of droplets, has been proposed by introducing the discrete SDF (DSDF). The DSDF is only valid for disperse phase and supposes that we can isolate droplets/bubbles in small volumes. However, it allows to describe droplets and bubbles with arbitrary shapes as long as they are homeomorphic to a sphere and this provides us with an interesting theoretical framework, which naturally degenerates to the previous work when the objects are spherical (chapter 4).

In a second main contribution, we have defined the spatially averaged SDF (ASDF), with an averaging kernel bounded to a small region around the interface and which preserves some information of the standard SDF given by the first moments of this distribution. We have shown that the ASDF degenerates to the DSDF, when the liquid or gas phase is disperse. In this case, the link with the NDF can be identified straightforwardly and explicitly. We have then shown how we can derive reduced-order models from an equation on the SDF using the moments of these distributions. However, we still need a closure modeling in situations of complex topological changes, while we have illustrated that in some simplified situations (spherical droplets), a closure model can be derived similar to the one given in 4.

Finally, to illustrate and assess the theoretical part, we have designed a new algorithm to extract the curvatures and the two different distributions NDF and SDF from a level-set field obtained with the DNS ARCHER code. This new algorithm preserves some geometrical and topological information, which essentially allows us to compute a NDF from an ASDF. This new tool will serve, in the next chapter and also in future work, to post-process more representative two-phase flows DNS simulations Canu et al.(2017), with and without topological changes, and thus propose closure modeling of the curvatures evolutions and possibly closure of the distribution from its moments.

Chapter 6

Methodology and modeling of separated and disperse phases

Remark 2 *By means of the present chapter, we will clarify our strategy to develop a unified model for both the disperse and separated phases. For this purpose, we work in collaboration with the CORIA laboratory to investigate some DNS two-phase flow computations that can provide a deep insight on the flow evolution and to model the transfer from resolved scales to sub-scales and vice-versa. The contribution of this chapter is still a work in progress, where we propose some new ideas and recommendations that need to be investigated in future works.*

6.1 Introduction

In the last two chapters, we have introduced a new approach to describe the separated and the disperse phases, based on some geometrical variables of the gas-liquid interface. The probabilistic approach given by the Surface Density Function (SDF) and its link with the NDF have allowed to describe, within a unified framework, the disperse and separated regimes of the two-phase flow. In the case of spherical droplets, we have managed to derive a closure model that involves averaged geometrical variables to describe the polydispersion of an evaporating spray. Nevertheless, we still have to further clarify our approach to design a unified closed model that can simulate the whole injection process.

Our modeling strategy consists in coupling a two-fluid model as the one proposed in Drui(2017) with a sub-scale model to simulate the flow in the whole domain. First, the two-fluid model solves the main interface structures in the separated phases. While the finest interface scales or the disperse phase will be modeled by an appropriate sub-scale approach. In the present work, we use the averaged geometrical quantities that we introduced in the last two chapters: the surface area density and the averaged mean and Gauss curvatures. These quantities are used to derive a sub-scale model for the unresolved interfaces and the disperse phase. In the separated phases, these quantities are expressed as moments of the SDF and they provide a statistical description of the interface topology, the ligaments and the drops. In the disperse phase, the same quantities have been expressed as the moments of the NDF to model the polydispersion. Another important variable is the volume fraction. The volume fraction of the disperse phase is expressed by using the droplet volume as an additional internal variable of the NDF. However, its expression in the separated phases has not yet been clarified. In the present chapter, we show how to express the four variables (volume fraction, surface area density and averaged mean and

Gauss curvatures) in the two regions of the flow. In the second part of this chapter, we classify the internal variables into resolve and unresolved scales, and then we discuss the interaction between a two-fluid model and a sub-scale model that we express as a system of equations on the averaged geometrical variables. Finally, the time evolution of the internal variables of the two distributions (SDF and NDF), as well as the stretching factor and the breakup rate will need to be simplified and closed. For this objective, we propose to investigate DNS computations that can provide more information on the gas-liquid interface evolution and to understand the mechanisms that controls this evolution.

6.2 Averaged geometrical quantities to describe disperse and separated phases

The Averaged SDF (ASDF) and the localized SDF, introduced in the last chapter, are two different distributions on the surface and are defined independently of the flow regime: separated or disperse phases. However, the two distributions share the same first order moments. The main objective of introducing the ASDF was to show a link between a statical description of the surface and the statistics on isolated objects. In the present chapter, the averaged or the localized SDF will be written in the following form:

$$F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}) = \left\langle \Sigma_h(t,\boldsymbol{x})\delta(\widehat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\boldsymbol{x}))\right\rangle, \tag{6.1}$$

where $\Sigma_h(t,\boldsymbol{x})$ is a surface area measure and $\boldsymbol{\xi}(t,\boldsymbol{x})$ is a vector of localized or averaged interfacial quantities: mean and Gauss curvatures and interface velocity. We underline that one can obtain the localized SDF from the averaged SDF by making the spatial averaging scale $h \to 0$ (see chapter 5). From now on, by abuse of notation, we simply call the different distributions on the surface by SDF and only when we need some higher level of precision to emphasize the differences between the distributions, we will use the proper notations.

Even though the SDF is defined independently of the flow regimes, the NDF is still the appropriate distribution to describe the sprays. Thus, we use the SDF to describe the separated phases, while the NDF is used once the spray is generated. Two sets of internal variables are considered: 1- For the SDF, we use the mean H and Gauss G curvatures and the interface velocity $\mathbf{v}_I$, 2- For the NDF, we use the surface area $S = 4\pi/\overline{G} > 0$, the mean radius $R = 1/(-\overline{H}) > 0$ and the mass velocity $\mathbf{v}_b$ of the droplet, where $\overline{\bullet}$ stands for the average over the droplets surface and defined in (5.29).

We summarize the statistical description of the two zones in Table 6.1.

6.2.1 Common moments of the SDF and the NDF

As explained in the previous two chapters, we reduce the evolution equation on the SDF or the NDF by deriving a system of equations on the moments of one of these distributions. One of the relevant question that needs further investigations is the number of moments needed to obtain a sufficient description of the flow features. Concerning the polydispersion of a cloud spherical droplets, it is clear that the minimum number of moments that allows to reconstruct a distribution with of a non-zero variance is three. In the EMSM model of Kah(2010), four integer size-moments have been used to capture the polydispersion. Therefore, we will need at least four moments (without counting the moments on the velocity) to obtain the same level of the description as the EMSM model for polydispersion. However, the number of moments

separated phases	disperse phase
$\boldsymbol{\xi_s} = (H, G, \mathbf{v}_I)$	$\boldsymbol{\xi_d} = (R, S, \mathbf{v}_b)$
$F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s}) = \left\langle \Sigma_h(t, \boldsymbol{x})\delta(\widehat{\boldsymbol{\xi}_s} - \boldsymbol{\xi}_s(t, \boldsymbol{x}))\right\rangle$	$f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_d}) = \left\langle \sum_k \delta(\boldsymbol{x} - \boldsymbol{x}_k(t))\delta(\widehat{\boldsymbol{\xi}_d} - \boldsymbol{\xi}_{dk}(t))\right\rangle$
$\underbrace{d_t(F)}_{\text{Transport}} + \underbrace{\partial_{\widehat{\boldsymbol{\xi}_s}} \cdot \left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c F\right)}_{\text{Interface evolution}} = \underbrace{\left\langle \dot{\mathcal{S}} \right\rangle_c F}_{\text{Surface stretching}}$	$\underbrace{d_t f}_{\text{Transport}} + \underbrace{\partial_{\widehat{\boldsymbol{\xi}_d}} \cdot \left(\left\langle \dot{\boldsymbol{\xi}}_d \right\rangle_c f\right)}_{\text{Droplet evolution}} = \underbrace{\langle \dot{\chi}/\chi \rangle_c f}_{\text{Topology variation}}$

Table 6.1: *A summary of the two descriptions of the disperse and the separated phases, line by line we have: the internal variables, the distributions and their transport equations. Where* $f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_d})$ *is the NDF,* $\dot{\mathcal{S}}$ *is the stretch rate factor,* χ *is the euler characteristic (see the last chapter for more details) and* $d_t(F) = \partial_t F + \mathbf{v}_I \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (F)$ *and* $d_t(f) = \partial_t f + \mathbf{v}_b \boldsymbol{\nabla}_{\boldsymbol{x}} \cdot (f)$ *are the Lagrangian-time derivatives following respectively the interface and the droplet velocity.*

separated phases	disperse phase
$\Sigma = \int_{\Omega_{\xi_s}} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s}) d^5\widehat{\boldsymbol{\xi}_s}$	$\Sigma_d = \int_{\Omega_{\xi_d}} \widehat{S} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_d}) d^5\widehat{\boldsymbol{\xi}_d}$
$\Sigma\widetilde{H}_s = \int_{\Omega_{\xi_s}} \widehat{H} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s}) d^5\widehat{\boldsymbol{\xi}_s}$	$\Sigma_d\widetilde{H}_d = \int_{\Omega_{\xi_d}} \widehat{R}^{-1}\widehat{S} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_d}) d^5\widehat{\boldsymbol{\xi}_d}$
$\Sigma\widetilde{G}_s = \int_{\Omega_{\xi_s}} \widehat{G} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s}) d^5\widehat{\boldsymbol{\xi}_s}$	$\Sigma_d\widetilde{G}_d = 4\pi \int_{\Omega_{\xi_d}} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_d}) d^5\widehat{\boldsymbol{\xi}_d}$

Table 6.2: *The surface area density and the averaged mean and Gauss curvatures defined as moments of the SDF or the NDF.*

to describe the topology in the separated phases is still an open question. In the last two chapters, four mean quantities have been identified to describe the gas-liquid interface and the polydispersion. These variables are: the surface area density, the averaged Gauss and mean curvatures and the volume fraction. The three first variables are common for both the disperse and the separated phases. We summarize their expressions in the following table: The averaged geometrical quantities given in Table 6.2 are the same quantities in the two regions of the flow, even if we are using two different distributions and internal variables. We can show this result by using Proposition 5.5.1. This shows the possibility to describe both regions using common variables.

6.2.2 The volumetric distribution and the volume fraction variable

In this section, we propose a solution to define correctly the volume fraction in the two regions of the flow. In the disperse phase, the natural way to obtain the volume fraction is to use the volume of the droplets as an additional internal variable, such that the phase space of the NDF becomes $\boldsymbol{\xi_d} = (R, S, v, \mathbf{v}_b)$ and v is the droplets volume. In this case, the volume fraction of the disperse phase reads:

$$\alpha_d = \int_{\widehat{\boldsymbol{\xi}_d}} \widehat{v} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_d}) d\widehat{\boldsymbol{\xi}_d}. \tag{6.2}$$

However, in the case of separated phases, we can not obtain a similar expression with the SDF. Indeed, the SDF is a distribution that measures the probable surface area having some given interfacial properties. As a matter of fact, it can not cope with the volume occupied by the liquid or the gas phase. We propose then to extend the definition of the SDF to be able to recover the volume fraction in a consistent way. For this purpose, we rely on the idea presented in Canu et al.(2017), where we use the distance function on the gas-liquid interface as an additional internal variable. The new distribution, contrary to the SDF, provides information on the volume occupied by each of the two phases, by giving the distribution of the flow with respect to the distance to the interface. We call this new distribution the Volumetric SDF (VSDF), which is defined here as follows:

$$F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}_s}) = \left\langle \delta(\widehat{\phi}-\phi(t,\boldsymbol{x}))\delta(\widehat{\boldsymbol{\xi}_s}-\boldsymbol{\xi}_s(t,\boldsymbol{x}))\right\rangle, \tag{6.3}$$

where the internal variables are composed of the distance function $\widehat{\phi}$, the Gauss and the mean curvatures and the interface velocity $\widehat{\boldsymbol{\xi}_s} = (\widehat{H},\widehat{G},\widehat{\mathbf{v}_I})$. But here, the vector $\boldsymbol{\xi}_s(t,\boldsymbol{x})$ is different from the one defined for the SDF. Indeed, the curvatures and the velocity at a point $\boldsymbol{x}$ are defined as the properties of the iso-surface level set that crosses this point, `i.e.` the curvatures at points $\boldsymbol{x}$ are the curvatures of the surface defined by the points $\boldsymbol{y}$ such that $\phi(t,\boldsymbol{y}) = \phi(t,\boldsymbol{x})$. Finally, the quantity $F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}}_s)d\boldsymbol{x}d\widehat{\phi}d\widehat{\boldsymbol{\xi}}_s$ measures the probable volume of the point $\boldsymbol{y}$ within a volume space $[\boldsymbol{x},\boldsymbol{x}+d\boldsymbol{x}]$ such that $\phi(t,\boldsymbol{y}) \in [\widehat{\phi},\widehat{\phi}+d\widehat{\phi}]$ and $\boldsymbol{\xi}_s(t,\boldsymbol{y}) \in [\widehat{\boldsymbol{\xi}}_s,\widehat{\boldsymbol{\xi}}_s+d\widehat{\boldsymbol{\xi}}_s]$. After this definition, different questions may arise. The first is the physical meaning of the level-set iso-surface and their associated surface properties: curvatures and velocity. Up to now, we have not investigated this new distribution. But, a similar distribution has been explored by Dumouchel et al.(2015) to introduce similarity scales concept in the atomization process. The second question concerns the relation between the SDF and the VSDF. We can derive such a link for the localized SDF. Indeed, the definition of the SDF is independent of the definition of the surface function $g(t,\boldsymbol{x})$ (see chapter 5). Therefore, we can take $g(t,\boldsymbol{x}) = \phi(t,\boldsymbol{x})$. In this case, the localized SDF can be expressed as follows:

$$\begin{aligned} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}}_s) &= \left\langle \delta\left(\phi(t,\boldsymbol{x})\right)\delta\left(\widehat{\boldsymbol{\xi}}_s-\boldsymbol{\xi}_s(t,\boldsymbol{x})\right)\right\rangle, \\ &= F_\phi(t,\boldsymbol{x};\widehat{\phi}=0,\widehat{\boldsymbol{\xi}}_s), \end{aligned} \tag{6.4}$$

since $\phi(t,\boldsymbol{x})$ is a distance function and $||\nabla_{\boldsymbol{x}}(\phi)|| = 1$.

Now, it is possible to express the volume fraction of the liquid by using the VSDF. Let us suppose that the positive level-set corresponds to the liquid phase. In this case, the liquid volume fraction can be expressed as follows:

$$\alpha(t,\boldsymbol{x}) = \int_{\widehat{\phi}\geq 0}\int_{\widehat{\boldsymbol{\xi}}_s} F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}}_s)d\widehat{\phi}d\widehat{\boldsymbol{\xi}}_s. \tag{6.5}$$

The surface area density and the two averaged curvatures are now a function of the distance function:

$$\begin{aligned} \Sigma(t,\boldsymbol{x};\widehat{\phi}) &= \int_{\widehat{\boldsymbol{\xi}}_s} F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}}_s)d\widehat{\boldsymbol{\xi}}_s \\ \Sigma\widetilde{H}(t,\boldsymbol{x};\widehat{\phi}) &= \int_{\widehat{\boldsymbol{\xi}}_s} \widehat{H}F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}}_s)d\widehat{\boldsymbol{\xi}}_s \\ \Sigma\widetilde{G}(t,\boldsymbol{x};\widehat{\phi}) &= \int_{\widehat{\boldsymbol{\xi}}_s} \widehat{G}F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}}_s)d\widehat{\boldsymbol{\xi}}_s \end{aligned} \tag{6.6}$$

These variables describe the average geometry of a level-set iso-surface at a given distance $\widehat{\phi}$ with respect to the gas-liquid interface. The averaged geometrical variables describing the gas-liquid interface is then given for $\widehat{\phi} = 0$.

6.3 Interface sub-scale modeling

6.3.1 Resolved and unresolved scales of a two fluid model

We recall that our strategy consists in using an appropriate two-fluid model to describe the large scales, while improving the description of the small scales by using moments of the NDF, of the SDF or of the VSDF. Close enough to the nozzle outlet, the interface is not so much deformed. A two-fluid model can then capture accurately the interface, provided that we are using a sufficient resolution. However, in the transition zone or in the disperse phase region, it is difficult to keep an accurate description of the real interface using a two-fluid model at an affordable computational cost. The interface will be diffused in a thick region. In this region, the volume fraction of the two-fluid model varies smoothly between 0 to 1, as it is illustrated in Figure 6.1. Some interface sharpen methods discussed in section 2.6 can be used to limit the artificial numerical diffusion. While the small scales compared to the grid resolution should be modeled.

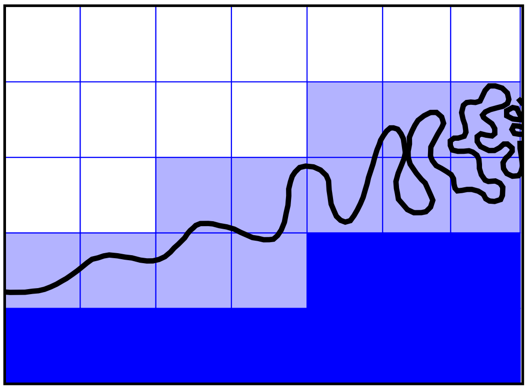


Figure 6.1: *Illustration of the interface diffusion of a two-fluid (or mixture) model, and sub-scale representation of the two-phase interface, reprinted from Drui(2017).*

By using the averaged geometrical quantities, we aim at modeling the sub-scale of the gas-liquid interface and the disperse phase. For this reason, it is important to introduce a threshold scale δ_s that separates the scales between resolved and unresolved ones. A scale l that verifies $l \geq \delta_s$ is a resolved scale with respect to the two-fluid resolution while $l < \delta_s$ is an unresolved scale. The definition of the threshold scale obviously depends on the grid resolution. But, its definition should be studied theoretically in order to define the required range of resolved scales that enable an accurate description of the mean flow, and also in order to define an adequate range of sub-scales that can have some "universal" feature laws to model. Another point that will also need to be clarified, is the relation between the threshold scale δ_s and the averaging scale h used to define the ASDF (see chapter 5 for more details). This question is not yet investigated. But, since the moments of the ASDF are dedicated to model the scales lower than δ_s, h should be at least smaller than δ_s. In future work, the threshold scale δ_s and the averaged scale h need to be clarified further and interlinked to the physical properties and the computational parameters.

In the following, we characterize the internal variables of the different distributions depending on the threshold scale δ_s. First, the two principal curvatures (see the definition in chapter 5) of a resolved interface satisfy:

$$\begin{aligned} \frac{1}{|\kappa_1|} &\geq \delta_s, \\ &\text{and} \\ \frac{1}{|\kappa_2|} &\geq \delta_s. \end{aligned} \tag{6.7}$$

These two inequalities can be expressed for the resolved mean and Gauss curvatures as follows:

$$\max\left(-\frac{1}{\delta_s}\left(2H+\frac{1}{\delta_s}\right),\frac{1}{\delta_s}\left(2H-\frac{1}{\delta_s}\right)\right) \leq G \quad \text{and} \quad \frac{1}{\delta_s} \geq |H|, \tag{6.8}$$

while unresolved curvatures satisfy the following conditions:

$$\min\left(-\frac{1}{\delta_s}\left(2H+\frac{1}{\delta_s}\right),\frac{1}{\delta_s}\left(2H-\frac{1}{\delta_s}\right)\right) < G \quad \text{or} \quad \frac{1}{\delta_s} < |H|, \tag{6.9}$$

with the condition $H^2 \geq G$, which must be maintained for both resolved and unresolved scales. Figure 6.2 illustrates the resolved and unresolved scales in the (G, H)-plan. We highlight the fact that these inequalities have been derived for the curvatures without the spatial averaging. However, in the present work, we heuristically suppose that the same separation of resolved and unresolved scales is valid for the spatially averaged curvatures. In the disperse phase, we consider that all the structures are unresolved by the two-fluid model. Then, we simply suppose that we have $R < \delta_s$, $S < 4\pi\delta_s^2$ and $v < 4\pi/3\delta_s^3$. Finally, for the VSDF, we only consider the points that are near the interface $|\phi| < \delta_s$.

6.3.2 The sub-scale geometrical moments

In the following, we use the superscripts r and u to denote respectively the resolved and unresolved variable. For example: $\Omega^u_{\xi_s}$ and $\Omega^r_{\xi_s}$ refer respectively to the part of unresolved and the resolved phase space of the SDF. Since the disperse phase is at an unresolved scale, the internal variables associated to the NDF will be simply denoted as follows: $\widehat{\boldsymbol{\xi_d}}$ for the internal variables and Ω_{ξ_d} for the phase space. Finally, by classifying the different internal variables, we can also separate the moments of the different distributions, used in the context of this chapter, into resolved and unresolved scale moments. These moments are summarized in Table 6.3.

We can show the following relations between the resolved and the unresolved moments:

$$\begin{aligned} \Sigma_s^r + \Sigma_s^u &= \Sigma_s, \\ \Sigma_s^r \widetilde{H}_s^r + \Sigma_s^u \widetilde{H}_s^u &= \Sigma_s \widetilde{H}_s, \\ \Sigma_s^r \widetilde{G}_s^r + \Sigma_s^u \widetilde{G}_s^u &= \Sigma_s \widetilde{G}_s. \end{aligned} \tag{6.10}$$

In the following, we call the unresolved moments the *sub-scale geometrical moments*. In table 6.3, we have defined the sub-scale geometrical moments according to the flow regimes: separated or disperse phases. But as we have already underlined, the definition used in the separated phases with the SDF is general and valid for both regions. In the disperse phase, we do not define the resolved moments because the disperse phase is always considered unresolved by the two-fluid model. Finally, the resolved volume fraction in the separated phases can be obtained as follows:

$$\alpha_s^r = \alpha^t - \alpha_s^u, \tag{6.11}$$

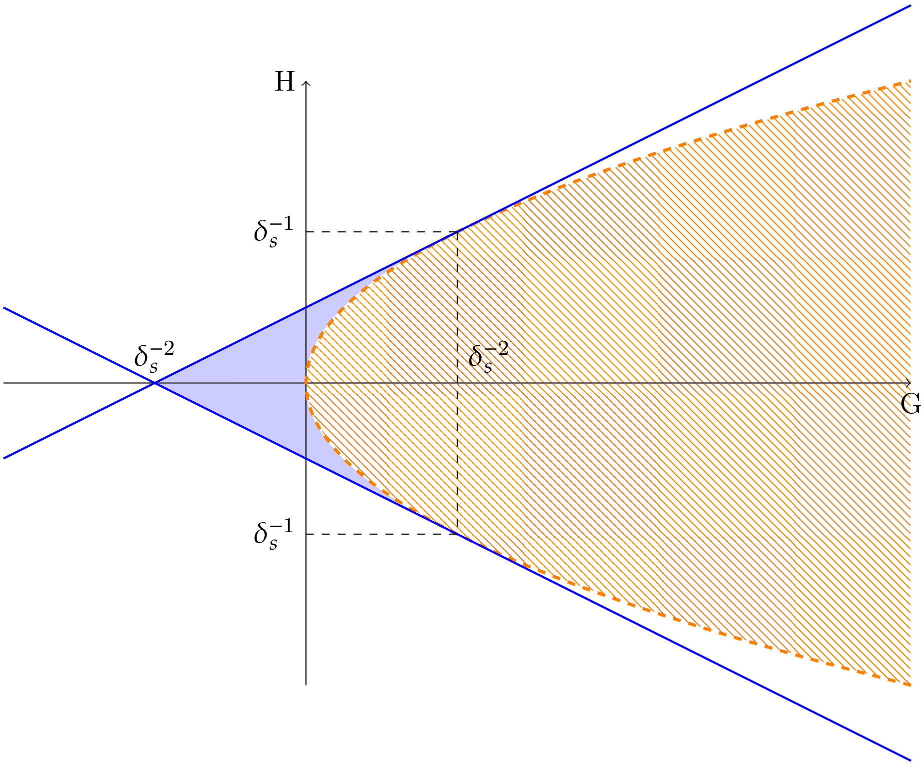


Figure 6.2: *(G, H)-plan: non-valid curvatures $H^2 < G$ (hashed region), resolved curvatures (blue region) and unresolved curvatures (the rest of the domain).*

	separated phases	disperse phase
Resolved scales	$\Sigma_s^r = \int\limits_{\Omega^r_{\xi_s}} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_s}}^r) d^5 \widehat{\boldsymbol{\xi_s}}^r$	not defined
	$\Sigma_s^r \widetilde{H}_s^r = \int\limits_{\boldsymbol{\xi_s}^r \in \Omega^r_{\xi_s}} \widehat{H}^r F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_s}}^r) d^5 \widehat{\boldsymbol{\xi_s}}^r$	not defined
	$\Sigma_s^r \widetilde{G}_s^r = \int\limits_{\Omega^r_{\xi_s}} \widehat{G}^r F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_s}}^r) d^5 \widehat{\boldsymbol{\xi_s}}^r$	not defined
Unresolved scales	$\Sigma^u = \int\limits_{\Omega^r_{\xi_s}} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_s}}^r) d^5 \widehat{\boldsymbol{\xi_s}}^r$	$\Sigma_d = \int_{\Omega_{\xi_d}} \widehat{S} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_d}}) d^5 \widehat{\boldsymbol{\xi_d}}$
	$\Sigma^u \widetilde{H}_s^u = \int\limits_{\Omega^r_{\xi_s}} \widehat{H} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_s}}^u) d^5 \widehat{\boldsymbol{\xi_s}}^u$	$\Sigma_d \widetilde{H}_d = \int\limits_{\Omega_{\xi_d}} \widehat{R}^{-1} \widehat{S} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_d}}) d^5 \widehat{\boldsymbol{\xi_d}}$
	$\Sigma \widetilde{H}_s^u = \int\limits_{\Omega^u_{\xi_s}} \widehat{G} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_d}}) d^5 \widehat{\boldsymbol{\xi_s}}^u$	$\Sigma_d \widetilde{G}_d = \int\limits_{\Omega_{\xi_d}} f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_d}}) d^5 \widehat{\boldsymbol{\xi_d}}$
	$\alpha_s^u = \int\limits_{\widehat{\phi}^r \in [0, \delta_s]} \int\limits_{\Omega^u_{\xi_s}} F_\phi(t, \boldsymbol{x}; \widehat{\phi}^u, \widehat{\boldsymbol{\xi_s}}^u) d^5 \widehat{\boldsymbol{\xi_s}}^u d\widehat{\phi}^u$	$\alpha_d = \int\limits_{\Omega_{\xi_d}} v f(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi_d}}) d^5 \widehat{\boldsymbol{\xi_d}}$

Table 6.3: *Separation of the moments into resolved and unresolved in separated and disperse phases.*

where α^t is the total volume fraction, the same as the one used in the two-fluid model.

After this separation of the scales, two different strategies can be envisioned to model the fuel injection process. First, one can couple a two-fluid model with two systems of equations on the geometrical variables: one for the resolved moments and an other one for the sub-scale geometrical moments. This idea is pretty similar to the one proposed in Devassy et al.(2015). Indeed, the authors have coupled the seven-equation two-fluid model and two systems of transport equations on the surface area density and the volume fraction: one associated to the separated phases and the second to the disperse phase. However, with this strategy, we can have some redundant information in our case. Indeed, the resolved volume fraction defined in 6.11 can be used to evaluate the curvatures associated to the resolved scales in as similar way as in Brackbill et al.(1992). The second strategy consists in coupling a two-fluid model with the system of equations on the sub-scale geometrical variables. As shown in table 6.3, these variables can have two representations: moments of the SDF and the VSDF or moments of the NDF. However, we underline again that these two sets of variables are the same. In modeling, it is often simple to consider the sub-scale as a population of non-spherical droplets rather than dealing with complex interface topology. While from an another point of view, such assumption leads to inaccurate modeling of the primary atomization. In future work, we will need to investigate carefully the different various strategies and to choose simplified but also physically valid assumptions. The final model should be able to predict the final polydispersion of droplets obtained from the primary and the secondary atomizations.

6.3.3 Sub-scale and two-fluid model interactions

In this section, we discuss the interactions between a sub-scale interface model and a two-fluid model. The sub-scale model is given by the dynamics of the sub-scale geometrical moments $(\alpha_s^u, \Sigma_s^u, \Sigma^u \widetilde{H}_s^u, \Sigma^u \widetilde{G}_s^u)$. The interactions between the two models should be expressed through exchange source terms. First, it is clear that the dynamics of the sub-scale geometrical moments are controlled by the dynamics of the mean scales of the flow. Indeed, the deformation and the atomization of the interface generate more smaller scales, which can not be captured by the two-fluid model. The two-fluid model behaves as it is diffusing the interface. This diffusion process can be explained by considering a statistical average over many realizations, where the uncertainties are very high. Consequently, this process generates sub-scales from the resolved scales. Meanwhile other phenomena such as the coalescence can have an opposite effect. We can derive the system of equations of the sub-scale geometrical moments from the evolution equations of the SDF and the VSDF. The obtained equations read as follows:

$$
\begin{aligned}
\partial_t \Sigma^u + \nabla_{\boldsymbol{x}} \cdot (\{\Sigma^u \, \overline{\mathbf{v}_I}\}) &= \int_{\widehat{\boldsymbol{\xi}_s}^{\,u}} < \dot{\mathcal{S}} >_c F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s}^{\,u}) \, d^5 \widehat{\boldsymbol{\xi}} + J_{r \Leftrightarrow u}^{\Sigma}, \\
\partial_t \Sigma^u \widetilde{H}_s^u + \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma^u \, \widetilde{H}_s^u \, \overline{\mathbf{v}_I}\right\}\right) &= \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma^u \widetilde{H}_s^u (\overline{\mathbf{v}_I} - \overline{\mathbf{v}_I}^H)\right\}\right) \\
&\quad + \int_{\widehat{\boldsymbol{\xi}_s}^{\,u}} < \dot{\mathcal{S}} + \dot{H}/\widehat{H} >_c \widehat{H} F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s{}^u}) \, d^5 \widehat{\boldsymbol{\xi}_s}^{\,u} + J_{r \Leftrightarrow u}^{\Sigma \widetilde{H}_s}, \\
\partial_t \Sigma \widetilde{G}_s^u + \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma^u \, \widetilde{G}_s^u \, \overline{\mathbf{v}_I}\right\}\right) &= \nabla_{\boldsymbol{x}} \cdot \left(\left\{\Sigma^u \widetilde{G}_s^u (\overline{\mathbf{v}_I} - \overline{\mathbf{v}_I}^G)\right\}\right), \\
&\quad + \int_{\widehat{\boldsymbol{\xi}_s}^{\,u}} < \dot{\mathcal{S}} + \dot{G}/\widehat{G} >_c \widehat{G}_s^u F(t, \boldsymbol{x}; \widehat{\boldsymbol{\xi}_s{}^u}) \, d^5 \widehat{\boldsymbol{\xi}_s}^{\,u} + J_{r \Leftrightarrow u}^{\Sigma \widetilde{G}_s}, \\
\partial_t \alpha_s^u + \nabla_{\boldsymbol{x}} \cdot (\{\alpha_s^u \, \overline{\mathbf{v}_I}\}) &= \nabla_{\boldsymbol{x}} \cdot (\{\alpha_s^u (\overline{\mathbf{v}_I} - \overline{\mathbf{v}_I}^\alpha)\}) \\
&\quad + \int_{\widehat{\phi} \in [0, \delta_s]} \int_{\widehat{\boldsymbol{\xi}_s}^{\,u}} < \dot{\mathcal{S}} >_c F_\phi(t, \boldsymbol{x}; \widehat{\phi}, \widehat{\boldsymbol{\xi}_s{}^u}) \, d^5 \widehat{\boldsymbol{\xi}_s}^{\,u} \, d\widehat{\phi} + J_{r \Leftrightarrow u}^{\alpha_s},
\end{aligned}
$$

(6.12)

where the different averaged velocities $(\overline{\mathbf{v}_I}, \overline{\mathbf{v}_I}^H, \overline{\mathbf{v}_I}^G, \overline{\mathbf{v}_I}^\alpha)$ are defined in (5.28) by substituting $\widehat{\boldsymbol{\xi}_s}$ by $\widehat{\boldsymbol{\xi}_s}^u$. This system of equations is slightly different from the one derived in (5.27) or in (5.45). Indeed, the integration over the internal variables is restricted to the unresolved phase space $\Omega^u_{\xi_s} \subset \Omega_{\xi_s}$. For this reason, we obtain two different parts of the source terms in the RHS of the equations. The first part is similar to the source terms obtained in (5.27) or (5.45). These source terms express the evolution of the internal variables within the unresolved phase space $\Omega^u_{\xi_s}$ and the stretching of the interface sub-scale part. The second type of source terms is related to the production and the destruction of the sub-scales from or to the resolved scales, i.e. the resolved scales become unresolved and vice versa. These source terms are expressed as follows:

$$\begin{aligned}
J^{\Sigma}_{r\Leftrightarrow u} &= \int\limits_{\partial\Omega^u_{\xi_s}} \left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c \cdot \boldsymbol{n}_{\xi_s}\right) F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}_s}) d\widehat{\boldsymbol{\xi}_s}, \\
J^{\Sigma\widetilde{H}}_{r\Leftrightarrow u} &= \int\limits_{\partial\Omega^u_{\xi_s}} \left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c \cdot \boldsymbol{n}_{\xi_s}\right) \widehat{H} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}_s}) d\widehat{\boldsymbol{\xi}_s}, \\
J^{\Sigma\widetilde{G}}_{r\Leftrightarrow u} &= \int\limits_{\partial\Omega^u_{\xi_s}} \left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c \cdot \boldsymbol{n}_{\xi_s}\right) \widehat{G} F(t,\boldsymbol{x};\widehat{\boldsymbol{\xi}_s}) d\widehat{\boldsymbol{\xi}_s}, \\
J^{\alpha}_{r\Leftrightarrow u} &= \int\limits_{\widehat{\phi}\in[0,\delta_s]} \int\limits_{\partial\Omega^u_{\xi_s}} \left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c \cdot \boldsymbol{n}_{\xi_s}\right) F_\phi(t,\boldsymbol{x};\widehat{\phi},\widehat{\boldsymbol{\xi}_s}) d\widehat{\boldsymbol{\xi}_s} d\widehat{\phi},
\end{aligned} \tag{6.13}$$

where $\partial\Omega^u_{\xi_s}$ is the boundary of the phase space $\Omega^u_{\xi_s}$ and $\boldsymbol{n}_{\xi_s}$ is the normal vector at the boundary set. These source terms can be separated into production terms when $\left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c \cdot \boldsymbol{n}_{\xi_s}\right) \geq 0$ and destruction terms when $\left(\left\langle \dot{\boldsymbol{\xi}}_s \right\rangle_c \cdot \boldsymbol{n}_{\xi_s}\right) < 0$. The production corresponds to the interface deformations and the primary atomization, while the destruction corresponds to the coalescence and the segregation of the small scales, nourishing larger scales. For this reason, the modeling of these source terms has to be coupled with the two-fluid model, especially for the modeling of the sub-scales production.

Let us now consider the effects of the unresolved scales on the two-fluid model, and how they may be accounted for. In Drui et al.(2016b), a two-fluid model is derived from Hamilton's variational principle following Gavrilyuk and Saurel(2002). Drui et al use a sub-scale energy interface pulsations as an additional kinetic energy in the Hamiltonian formulation. Furthermore, a sub-scale dissipation was associated to the pulsation kinetic energy. The obtained model depends on two parameters, interpreted as a micro-inertia and a micro-viscosity associated to the gas-liquid interface sub-scale. By considering a monodisperse bubbly flow, the micro-inertia and the micro-viscosity of the bubble pulsations have been identified and related to the bubble radius and other physical variables of the flow. The effects of these sub-scale features on the two-fluid model were identified by investigating their effects on the dispersion relations of the sound speed. The sub-scale geometrical moments should allow a more accurate description of the interface sub-scale. It will be then important to benefit from this advantage to include more physical effects in the two-fluid model. For example, a surface potential energy could be written as a function of these quantities. In the same way, the pulsation energy could apply to deformed interface and then, the volume variations would be functions of the geometrical variables. Departing from these energies and using the procedure based on Hamilton's variational principle and the second principle of thermodynamics, as explained in Drui(2017), we may enrich again the two-phase models. These works are the subject of two PhDs Cordesse(2019) and Di Battista(2020). The first one aims at enriching the two-fluid model with more disequilibrium between the two phases, while ensuring a well-posed mathematical properties. The second aims at enriching the two-phase models using the Hamilton's variational principle that includes the sub-scale geometrical moments.

6.3.4 A simplified model for the unresolved scales

In the previous section, we wrote a general form of a system of equations satisfied by the sub-scale geometrical moments. The source terms involved in this system need to be closed by proposing an evolution of the curvatures and the stretching factor, as well as a proper reconstruction of the SDF and of the VSDF. In the following, we show how to close the model in a simplified case. For this purpose, we consider the following assumptions:

- the sub-scales are represented by a disperse phase and the sub-scale geometrical moments will be expressed as moments of the NDF $f(t,\boldsymbol{x};\widehat{R},\widehat{S},\widehat{v},\widehat{\mathbf{v}_b})$,
- we use the mixture two-fluid model developed in Drui(2017) and we suppose a very dilute disperse phase $\alpha << 1$. Consequently, this mixture model is used to simulate the continuous gaseous phases (by assuming a negligible volume fraction in the two-fluid model). The only interaction that we consider between the gaseous phase and the disperse phase is due to the exchange in momentum through the drag force exerted by the gas on a droplet. The drag force is modeled using the Stokes law:

  $$\boldsymbol{F} = \frac{\boldsymbol{u}_g - \mathbf{v}_b}{\theta S}, \tag{6.14}$$

  where $\theta = \dfrac{18\pi\mu_g}{\rho_l}$ is a constant coefficient.
- we assume a monokinetic assumption for the droplet velocity:

  $$f(t,\boldsymbol{x};\widehat{R},\widehat{S},\widehat{v},\widehat{\mathbf{v}_b}) = N(t,\boldsymbol{x};\widehat{R},\widehat{S},\widehat{v})\,\delta(\widehat{\mathbf{v}_b} - \boldsymbol{u}_p(t,\boldsymbol{x})), \tag{6.15}$$

- we suppose that at a given position and time all the droplets can be parametrized with one size variable and we choose here the surface area $\widehat{S}$:

  $$N(t,\boldsymbol{x};\widehat{R},\widehat{S},\widehat{v}) = n(t,\boldsymbol{x};\widehat{S})\,\delta(\widehat{R} - \overline{R}(t,\boldsymbol{x};\widehat{S}))\,\delta(\widehat{v} - \overline{v}(t,\boldsymbol{x};\widehat{S})), \tag{6.16}$$

  where $\overline{R}(t,\boldsymbol{x};\widehat{S}) = 1/(2\sqrt{\pi})a(t,\boldsymbol{x})\widehat{S}^{1/2}$ and $\overline{v}(t,\boldsymbol{x};\widehat{S}) = 1/(6\sqrt{\pi}b(t,\boldsymbol{x})\widehat{S}^{3/2}$. The two scalars $a(t,\boldsymbol{x})$ and $b(t,\boldsymbol{x})$ will be related to the moments of the NDF. We can also remark that for $a(t,\boldsymbol{x}) = 1.$ and $b(t,\boldsymbol{x}) = 1.$, we obtain the condition of spherical droplets and the averaged geometrical quantities can be expressed again as fractional moments of $n(t,\boldsymbol{x};\widehat{S})$, as presented in chapter 4,
- we neglect the source terms that generate the sub-scale quantities from the resolved scales, i.e. $J^{\psi}_{r\Leftrightarrow u} = 0$, where $\psi = \alpha, \Sigma, \Sigma\widetilde{H}, \Sigma\widetilde{G}$,

The volume fraction, the surface area density and the two averaged mean and Gauss curvatures can be written as a function of the fractional size moments of the reduced distribution $n(t,\boldsymbol{x};S)$ and the two variables $a(t,\boldsymbol{x})$ and $b(t,\boldsymbol{x})$,as follows:

$$\begin{array}{rcl}
\Sigma_d\widetilde{G}_d(t,\boldsymbol{x}) & = & 4\pi\mathrm{m}_0(t,\boldsymbol{x}), \\
\Sigma_d\widetilde{H}_d(t,\boldsymbol{x}) & = & \dfrac{1}{2\sqrt{\pi}}a(t,\boldsymbol{x})\mathrm{m}_{1/2}(t,\boldsymbol{x}), \\
\Sigma_d(t,\boldsymbol{x}) & = & \mathrm{m}_1(t,\boldsymbol{x}), \\
\alpha_d(t,\boldsymbol{x}) & = & \dfrac{1}{6\sqrt{\pi}}b(t,\boldsymbol{x})\mathrm{m}_{3/2}(t,\boldsymbol{x}),
\end{array} \tag{6.17}$$

where $\mathrm{m}_{k/2} = \int_0^{S_{max}} S^{k/2} n(t, \boldsymbol{x}; S) dS$ and $S_{max} = 4\pi\delta_s^2$. To have closure relations between the four sub-scale geometrical moments and the two variables $a(t, \boldsymbol{x})$ and $b(t, \boldsymbol{x})$, we transport the following six quantities $(\Sigma_d \widetilde{G}_d, \Sigma_d \widetilde{H}_d, \Sigma_d, \alpha_d, \mathrm{m}_{1/2}, \mathrm{m}_{3/2})$. The system of equations of these quantities can be derived from the evolution equation of the NDF given in Table 6.1. The obtained equations read as follows:

$$
\begin{aligned}
\partial_t \Sigma_d \widetilde{G}_d + div(\Sigma_d \widetilde{G}_d \boldsymbol{u}_p) &= 4\pi \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} \right\rangle_c n(t, \boldsymbol{x}, S) dS, \\
\partial_t \Sigma \widetilde{H} + div(\widetilde{H} \boldsymbol{u}_p) &= \frac{1}{2\sqrt{\pi}} a(t, \boldsymbol{x}) \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} + \frac{\dot{S}}{S} + \frac{\dot{H}}{H} \right\rangle_c S^{1/2} n(t, \boldsymbol{x}, S) dS, \\
\partial_t \Sigma + div(\Sigma \boldsymbol{u}_p) &= \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} + \frac{\dot{S}}{S} \right\rangle_c S n(t, \boldsymbol{x}, S) dS, \\
\partial_t \alpha_d + div(\alpha_d \boldsymbol{u}_p) &= \frac{1}{6\sqrt{\pi}} b(t, \boldsymbol{x}) \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} + \frac{\dot{v}}{v} \right\rangle_c S^{3/2} n(t, \boldsymbol{x}, S) dS, \\
\partial_t \mathrm{m}_{1/2} + div(\mathrm{m}_{1/2} \boldsymbol{u}_p) &= \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} + 1/2 \frac{\dot{S}}{S} \right\rangle_c S^{1/2} n(t, \boldsymbol{x}, S) dS, \\
\partial_t \mathrm{m}_{3/2} + div(\mathrm{m}_{3/2} \boldsymbol{u}_p) &= \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} + 3/2 \frac{\dot{S}}{S} \right\rangle_c S^{3/2} n(t, \boldsymbol{x}, S) dS, \\
\partial_t \Sigma \boldsymbol{u}_p + div(\Sigma \boldsymbol{u}_p \otimes \boldsymbol{u}_p) &= \int_0^{S_{max}} \left\langle \frac{\dot{\chi}}{\chi} + \frac{\dot{S}}{S} \right\rangle_c S \boldsymbol{u}_p n(t, \boldsymbol{x}, S) dS + \mathrm{m}_0 \frac{\boldsymbol{u}_g - \boldsymbol{u}_p}{\theta}.
\end{aligned}
\tag{6.18}
$$

The source terms in the RHS of the equations depend on the time evolution of the internal variables that are given by $\langle \dot{v} \rangle_c$, $\left\langle \dot{R} \right\rangle_c$ and $\left\langle \dot{S} \right\rangle_c$, as well as the breakup rate of the droplets expressed as the time derivative of the logarithm of the Euler characteristic (we suppose the droplets are homeomorph to spheres) $\langle \dot{\chi}/\chi \rangle_c$. In the present work, we consider a simplify modeling of the secondary atomization inspired from the wave breakup model Reitz(1987):

A. Wave model assumes that the time of breakup and the resulting droplet size are related to the fastest-growing Kelvin-Helmholtz instability. The maximum growth-rate of a small perturbation on the interface corresponds to a frequency w_{bu} and a wave number λ_{bu} that are expressed as function of the Weber and Ohnesorge numbers as follows:

$$
\begin{aligned}
w_{bu} &= \frac{\left(0.34 + 0.38 We_g^{1.5}\right)}{(1 + Oh)(1 + 1.4(Oh\sqrt{We_g})^{0.6}} \left(\frac{\sigma}{\rho_l r^3}\right)^{0.5}, \\
\lambda_{bu} &= \frac{9.02\left(1 + 0.45 Oh^{0.5}\right)\left(1. + 0.4\left(Oh\sqrt{We_g}\right)\right)}{1 + 0.81 We_g^{1.67}} r,
\end{aligned}
\tag{6.19}
$$

where $We_g = \dfrac{\rho_l ||\boldsymbol{u}_p(t, \boldsymbol{x}) - \boldsymbol{u}_g(t, \boldsymbol{x})|| r}{\sigma}$ and $Oh = \dfrac{\mu_l}{\rho_l r \sigma}$. the variables ρ_l and μ_l are respectively the mass density and the viscosity of the liquid, σ is the surface tension and $r = \sqrt{S}$ is an approximate radius for a droplet of surface area S. Thus, the breakup time is given as follows:

$$
\tau_{bu}(We_g, Oh) = 3.76 B_1 \frac{r}{\lambda_{bu} w_{bu}}.
\tag{6.20}
$$

Finally, we use the breakup time to model the breakup rate:

$$
\left\langle \frac{\dot{\chi}}{\chi} \right\rangle_c = \begin{cases} \tau_{bu}^{-1}(We_g, Oh), & \text{if } \sqrt{S} > B_0 \lambda_{bu}, \\ 0, & \text{otherwise,} \end{cases}
$$

where B_0 and B_1 are two parameters of the wave breakup model.

B. We suppose incompressible droplets and each fragmentation of a droplet generates new droplets with the same volume:

$$\langle \dot{v} \rangle_c = \begin{cases} -\tau_{bu}^{-1} \widehat{v}, & \text{if } \sqrt{\widehat{S}} > B_0 \lambda_{bu}, \\ 0, & \text{otherwise.} \end{cases}$$

C. Finally, for the evolution of the surface area and the mean radius, we consider each breakup is followed by a relaxation toward a spherical form of the droplets:

$$\left\langle \dot{S} \right\rangle_c = \begin{cases} \tau_{bu}^{-1} \left(6\sqrt{\pi} \widehat{v}^{2/3} - \widehat{S} \right), & \text{if } \sqrt{\widehat{S}} > B_0 \lambda_{bu}, \\ 0, & \text{otherwise,} \end{cases}$$

and

$$\left\langle \dot{R} \right\rangle_c = \begin{cases} \tau_{bu}^{-1} \left(\left(\frac{3}{4\pi} \widehat{v} \right)^{1/3} - \widehat{R} \right), & \text{if } \sqrt{\widehat{S}} > B_0 \lambda_{bu}, \\ 0, & \text{otherwise.} \end{cases}$$

These relations close the model in the chosen simplified framework.

Some elements on the numerical resolution: The numerical resolution of system (6.18) is achieved through an operator splitting Doisneau et al.(2014); Descombes et al.(2014), where the transport part (LHS of the equations) and the source terms (RHS of the equations) are solved independently. The numerical resolution of the transport part is similar to the one of the fractional moments (4.24) and the difficulties to comply with the realizability issues are the same for the two models. The reader can refer to chapter 7, where robust and accurate numerical schemes are proposed for the numerical resolution of the fractional moment model in the case of spherical droplets. For the numerical resolution of source terms, we use the DQMOM approach. In this case, the distribution in size $n(t; S)$ (the space is no more considered in the source terms resolution) is first approximated by a sum of Dirac distributions:

$$n(t; S) = \sum_{l=1,2} n_l(t) \delta(S - S_l(t)), \tag{6.21}$$

such that the abscissas $S_l(t)$ and the weights $n_l(t)$ are related to the fractional moments of the distribution $n(t; S)$ as follows:

$$m_{k/2}(t) = \sum_{l=1,2} n_l(t) S_l^{k/2}(t), \tag{6.22}$$

where $k = 0, \ldots, 3$. In the DQMOM approach, we solve the evolution of the abscissas and the weights due to source terms. Using the evolution equation of the NDF, we can show that these quantities satisfy the following ODE system for $l = 1, 2$:

$$\begin{aligned} \frac{dn_l}{dt}(t) &= \tau_{bu,l}^{-1} n_l(t), \\ \frac{dS_l}{dt}(t) &= \tau_{bu,l}^{-1} \left(6\sqrt{\pi} v_l(t)^{2/3} - S_l(t) \right), \\ \frac{dR_l}{dt}(t) &= \tau_{bu,l}^{-1} \left(\left(\frac{3}{4\pi} v_l(t) \right)^{1/3} - R_l(t) \right), \\ \frac{dv_l}{dt}(t) &= -\tau_{bu,l}^{-1} v_l(t), \end{aligned} \tag{6.23}$$

where $\tau_{bu,l}$ is the breakup time (6.20) computed for each quadrature abscissa. The abscissas (surface areas) $S_l(t_n)$, at the nth iteration, are computed by using the PD algorithm Gordon(1968) which solves the system (6.22) and the initial volumes and radii are computed as follows:

$$\begin{aligned} R_l(t_n) &= \frac{1}{2\sqrt{\pi}}a(t_n)S_l^{1/2}(t_n), \\ v_l(t_n) &= \frac{1}{6\sqrt{\pi}}b(t_n)S_l^{3/2}(t_n). \end{aligned} \tag{6.24}$$

The time integration of this ODE in $[t_n, t_{n+1}]$ allows to approximate the solution of the transported variables at $t = t_{n+1}$ as follows:

$$\begin{aligned} \mathrm{m}_{k/2}(t_{n+1}) &= \sum_{l=1,2} n_l(t_{n+1})S_l^{k/2}(t_{n+1}), \\ \Sigma_d\widetilde{H}_d(t_{n+1}) &= \sum_{l=1,2} n_l(t_{n+1})S_l(t_{n+1})R_l^{-1}(t_{n+1}), \\ \alpha_d(t_{n+1}) &= \sum_{l=1,2} n_l(t_{n+1})v_l(t_{n+1}), \end{aligned} \tag{6.25}$$

2D jet of non-spherical droplets: In the following, we consider a 2D rectangular domain of size $[-0.5, 0.5] \times [-0.5, 0.5]$ (we use dimensionless variables). On the top ($y = 0.5$), bottom ($y = -0.5$) and right ($x = 0.5$) faces, we use the Neumann boundary conditions. On the left face ($x = -0.5$), we consider a small hole in the middle. The hole is of radius $\eta_{inj} = 1.e-2$. From the hole outlet, we inject droplets of uniform size distribution:

$$n(t, y, S) = 1, \tag{6.26}$$

where $S \in [0, 1]$ and $-\eta_{inj} < y < \eta_{inj}$. The initial droplets are not spherical with: $a = 0.7$ and $b = 0.5$. The droplets are injected with the following velocity profile at $x = -0.5$:

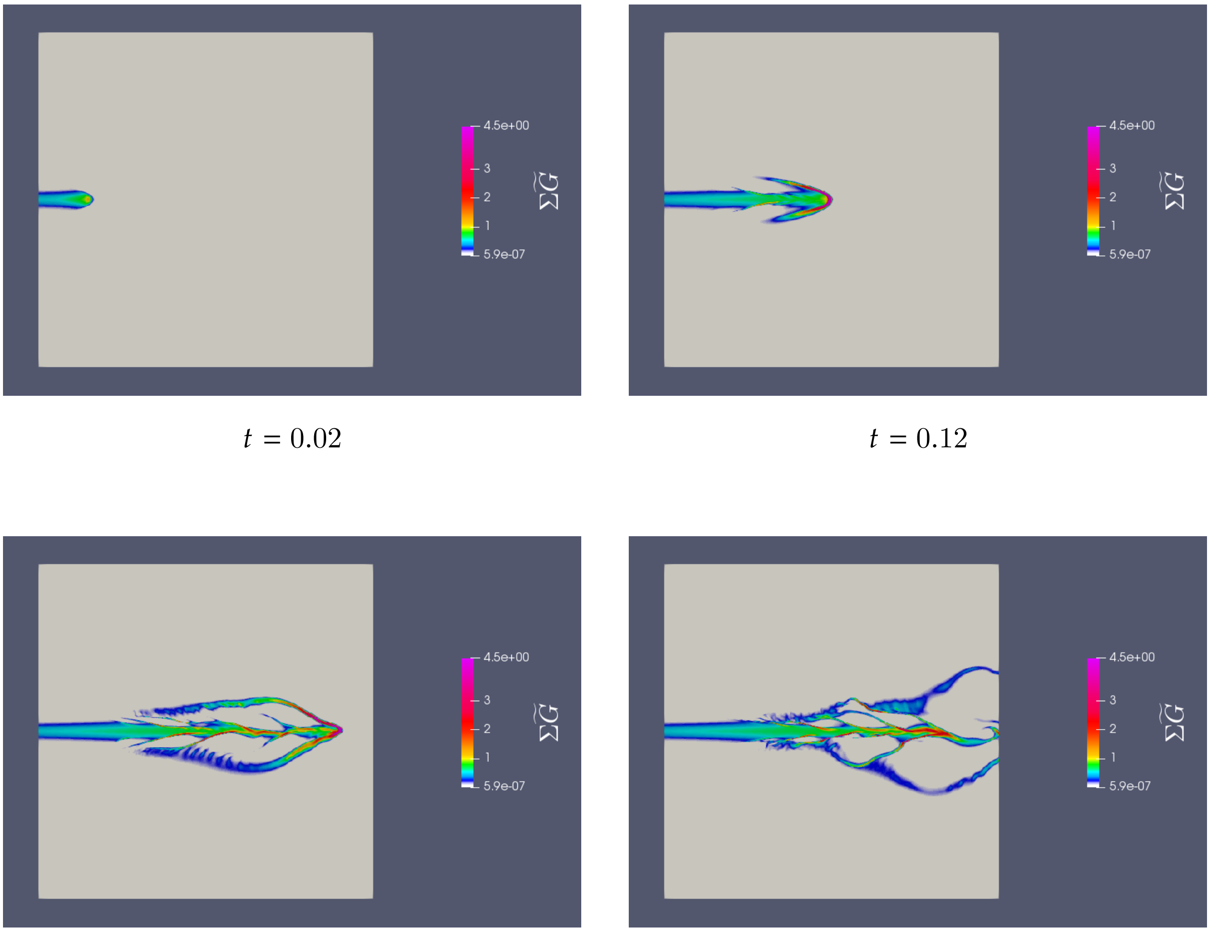
$$\begin{aligned} \boldsymbol{e}_x \cdot \mathbf{v}_b(t, y) &= 5\cos^4(\pi\frac{y}{2\eta_{inj}}), \\ \boldsymbol{e}_y \cdot \mathbf{v}_b(t, y) &= 0. \end{aligned} \tag{6.27}$$

The gas of a mass density $\rho_g = 1$ is also injected with following velocity field:

$$\begin{aligned} \boldsymbol{e}_x \cdot \boldsymbol{u}_g(t, y) &= 8\cos^4(\pi\frac{y}{2\eta_{inj}}), \\ \boldsymbol{e}_y \cdot \boldsymbol{u}_g(t, y) &= 0. \end{aligned} \tag{6.28}$$

The injection velocities of the gas and of the droplets are different. It leads to a strong relative velocity between the two phases and then to the fragmentation of the droplets. The numerical simulation is performed in Adaptive Mesh Refinement framework using the `CanoP` code, which will be presented later in chapter 9. The minimum cell size of the mesh is $\Delta x_{min} = 2^{-9}$ and the maximum size is $\Delta x_{max} = 2^{-7}$. Figures 6.3 and 6.4 illustrate respectively the number density ($\Sigma\widetilde{G}_d$) and surface are density at four different instants. We can remark the fragmentation by observing a global increase of number density and surface area density over time.

We recall that this simulation and the used model do not represent any realistic configuration. The coupling with the two-fluid model has not yet been completely achieved. Indeed, we used the mixture model of Drui(2017) to simulate the gas flow by assuming a negligible volume fraction. The objective of this first work is to illustrate the possibility of modeling the sub-scales with non-spherical droplets and takes into account the fragmentation and the two-way coupling

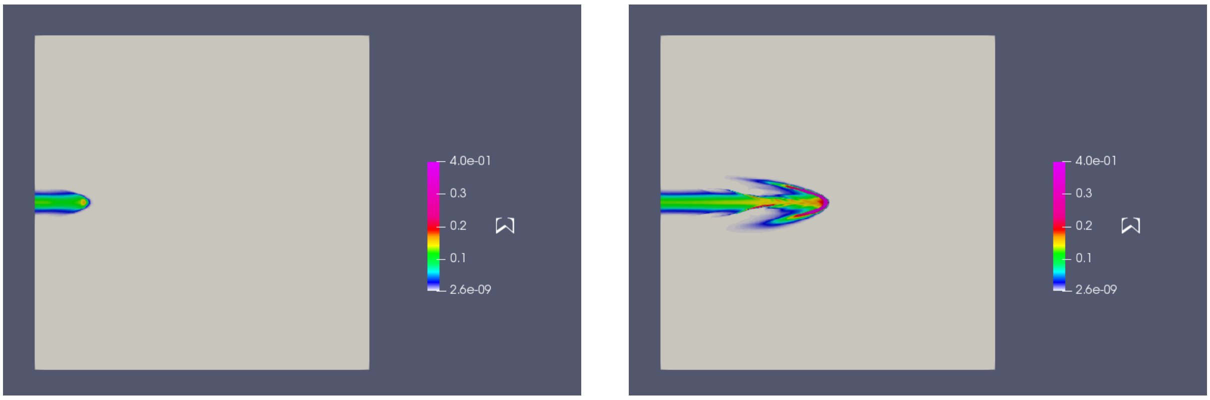


$t = 0.02$ $t = 0.12$

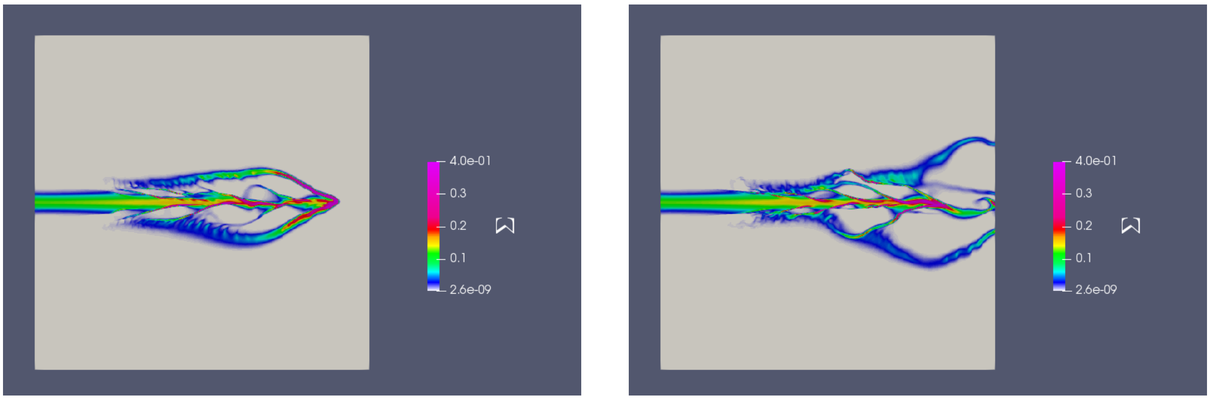
$t = 0.24$ $t = 0.38$

Figure 6.3: *Number density $\Sigma\widetilde{G}$ of the droplets given at four times for the injection of non-spherical droplets.*

with the gas. This model, given by the equations (6.18), can be considered as an extension of the fractional moment model of spherical droplets given in (4.24) to non-spherical droplets. The parametrization of the internal variables by the two parameters $a(t,\boldsymbol{x})$ and $b(t,\boldsymbol{x})$ has allowed to simplify the treatment of the phase space and the closure of the NDF. However, these parameterizations are not necessary the good ones. Indeed, when we have very small droplets $S << 1$, the droplet shape is more likely to be spherical in this case $a \simeq 1$ and $b \simeq 1$. While in the case of a large drop, its shape is more likely to be very different than a sphere and $a < 1$ and $b < 1$. Therefore, we will need to improve this parametrization and propose a more relevant one. Finally, this is a first test of our approach to the interface sub-scale by this geometrical variables. In future work, more realistic modeling will be assisted by investigating DNS results.



t = 0.02 t = 0.12



t = 0.24 t = 0.38

Figure 6.4: *Surface area density* Σ *of thedroplets given at four times for the injection of non-spherical droplets.*

6.4 DNS post-processing toward a modeling of the sub-scales

In the previous section, we explained our strategy to couple a two-fluid model with a sub-scale model based on the sub-scale geometrical moments. The sub-scale model depends on the time evolution of the internal variables of the SDF or of the NDF as well as the surface stretching and the breakup source terms. For future work, these terms will need to be correctly modeled and related to the transport variables of the flow. The main goal of this section is to extract the evolution of such geometrical properties from DNS test cases. The well-known Archer code Menard et al.(2007); Vaudor et al.(2017), where a combined VOF and Level-set approach is used to capture the interface and a ghost-fluid method is applied to represent accurately the jump of the variables across the liquid-gas interface, has been used to this end. We also use the algorithm presented in chapter 5 to extract the curvatures and to compute the two distributions: the SDF and the NDF. In this section, we present two numerical test cases:

- **Rayleigh-Plateau** test-case is presented to characterize the evolution of the interface curvatures in a simple breakup case, where a liquid column is split into several droplets
- **Liquid jet** test-case is a primary break-up simulation of an injected liquid. In this simulation, we show that the distribution of the curvatures on the interfaces can provide us with relevant information on the flow topology.

6.4.1 Plateau-Rayleigh simulation

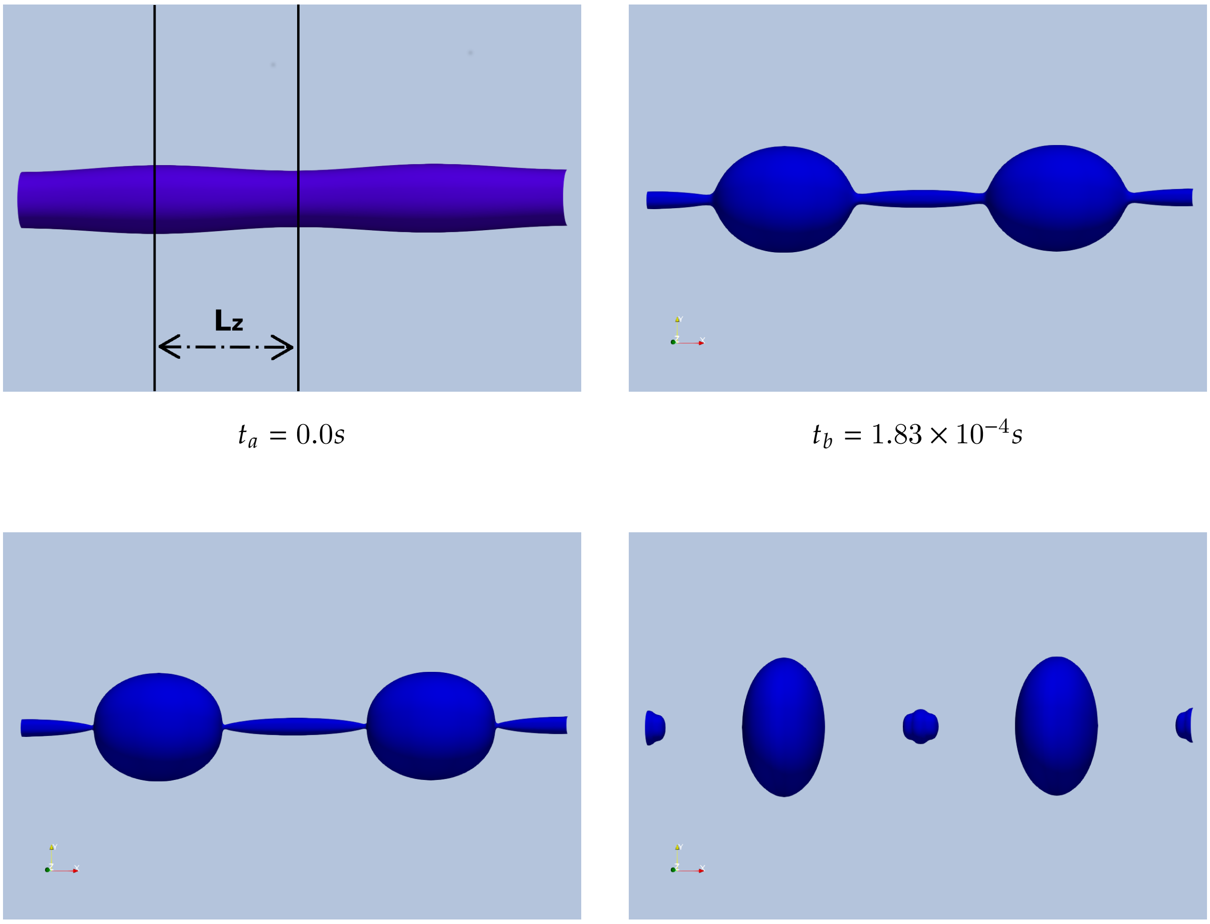
We consider an initial cylindrical column of liquid with a small sinusoidal deformation at the interface. The initial level-set function is given in cylindrical coordinates as follows:

$$\phi(t=0; r, \theta, z) = (r-R)(1+\epsilon \sin(2\pi k_z z)), \tag{6.29}$$

where k_z is a wave number satisfying $k_z R = 0.7$, $R = 3.34 \times 10^{-5} m$ being the radius of the cylinder and $\epsilon = 0.1$. Using the symmetry properties of this case, we only simulate a quarter of the cylinder, where the symmetric boundary conditions have been employed in a computational box $lx \times ly \times lz$ where $lx = ly = 1. \times 10^{-4} m$ and $lz = 1.5 \times 10^{-4} m$. We discretize the domain into $64 \times 64 \times 96$ Cartesian elements. The liquid and the gas properties are reported in table 6.4. In figure 6.5, four time frames have been selected as in Canu et al.(2017) to show the break-up process, which is conducted by the Plateau-Rayleigh instability. The four instants correspond to the initial configuration at $t_a = 0.s$, the deformation of the cylinder with the generation of the bottle-neck at $t_b = 1.83 \times 10^{-4} s$, the break-up moment at $t_c = 1.86 \times 10^{-4} s$ and the final generation of droplets at $t_d = 2.07 \times 10^{-4} s$. To have a deeper insight in the evolution of the curvatures, we chose these four instants to extract the mean and Gauss curvatures at the interface defined by $\phi(t; \boldsymbol{x}) = 0$. The spatial averaging of the curvatures presented in the last chapter is used here. In fact, the spatial averaging permits to obtain a structured and a clear picture of this evolution, compared to the case without averaging, where a large amount of noise is present, as illustrated in Figure 6.6. In Figure 6.7, we display the spatially averaged curvatures in the (G, H)-plan using a spatial averaging scale $h = 6\Delta x$, where Δx is the mesh

ρ_l kg.m^{-3}	μ_l kg.m^{-1}.s^{-1}	ρ_g kg.m^{-3}	μ_g kg.m^{-1}.s^{-1}	σ kg.s^{-2}
1000	$1,0.10^{-3}$	1.0	$1,879.10^{-3}$	$0,072$

Table 6.4: *Physical properties of the Rayleigh-Plateau simulation.*



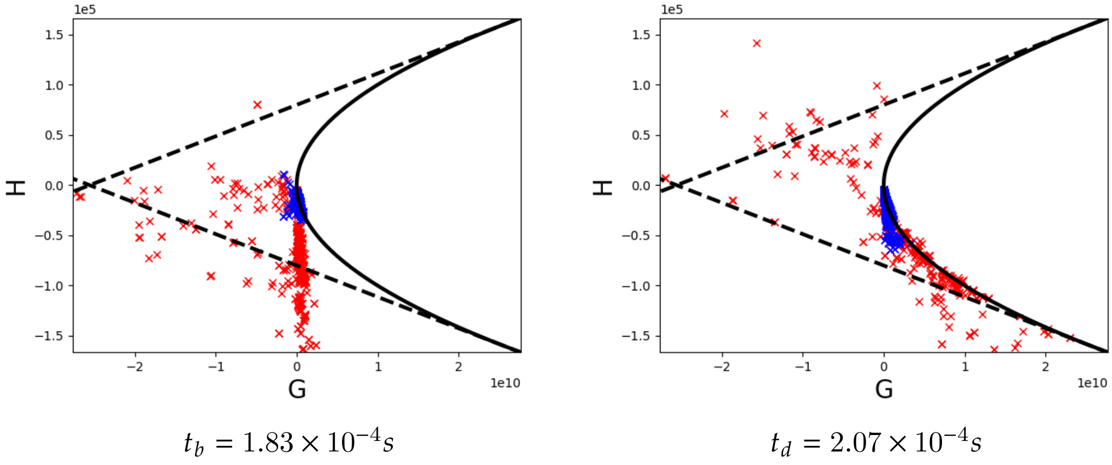
$t_a = 0.0s$ $t_b = 1.83 \times 10^{-4}s$

$t_c = 1.86 \times 10^{-4}s$ $t_d = 2.07 \times 10^{-4}s$

Figure 6.5: *The gas-liquid interface of Plateau-Rayleigh simulation. The two solid lines in the left-top corner determine the domain of the simulation.*

size. In the same figure, we plot the two lines limiting the resolved and the unresolved scales with a threshold scale $\delta_s = 3dx$. At the initial time $t = t_a$, the cylinder is weakly deformed and all the curvature points in the (G, H)-plan are all gathered around the vertical line at $G = 0$. At $t = t_b$, the cylinder is sufficiently deformed and the curvatures form an arc in (G, H)-plan. We also remark negative values of the Gauss curvature (top-right of Figure 6.7) that corresponds to the saddle points in the middle of the cylinder. In the bottom-left corner of Figure 6.7, which corresponds to the break-up time $t = t_c$, the arc formed in the previous step breaks up and two separated clouds of points can be identified. At the final time $t = t_d$, we obtain two droplets of approximately spherical shapes and of two different sizes. In bottom-right of Figure 6.7, two cloud of points are positioned close to the curve $G = H^2$. The one with the highest curvatures corresponds to the smallest droplet and the other to the biggest droplet.

The evolution in the (G, H)-plan reveals relevant information on the evolution of the gas-liquid interface. For example, a slow evolution toward high curvatures can reveal the development of some initial instabilities, while fast evolution accompanied with high curvatures can be related to the variation on the interface topology. Plateau-Rayleigh instabilities represent one of the breakup mechanisms of realistic injection configurations. It is important to classify the different mechanisms and to choose representative configurations as the one proposed here to understand



$t_b = 1.83 \times 10^{-4} s$

$t_d = 2.07 \times 10^{-4} s$

Figure 6.6: *Gauss and mean curvatures of the gas-liquid interface without spatial averaging at two instants: left ($t = t_b$) and right ($t = t_d$). The red points correspond to left part of*

ρ_l kg.m^{-3}	μ_l kg.m^{-1}.s^{-1}	ρ_g kg.m^{-3}	μ_g kg.m^{-1}.s^{-1}	σ kg.s^{-2}
776	4.0×10^{-3}	1.3	$1,78 \times 10^{-3}$	0.072

Table 6.5: *Physical properties of the gas and the liquid used in the jet flow simulation.*

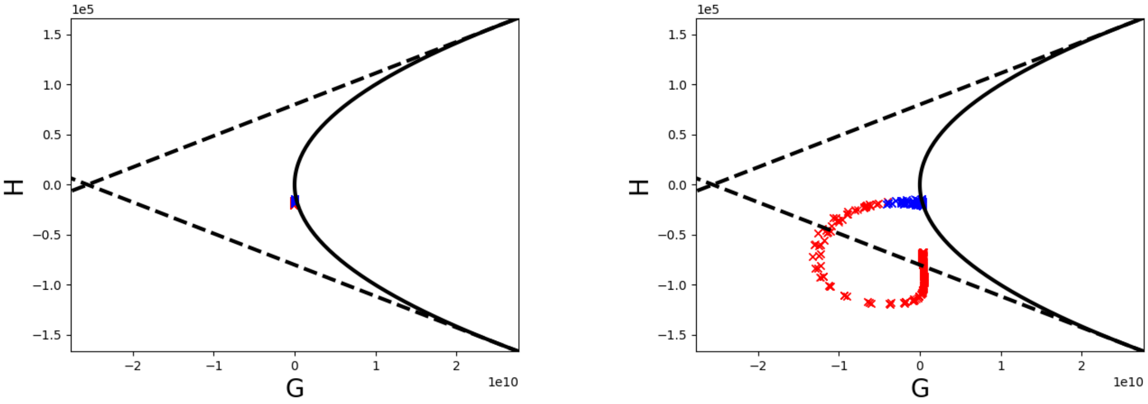
and model the evolution of the curvatures depending on each mechanisms.

6.4.2 Jet simulation

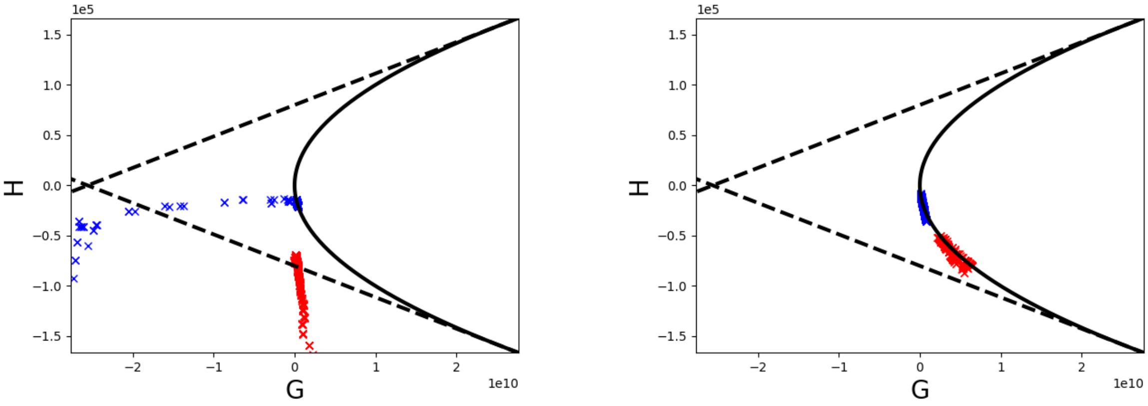
In this section, we present some first results of a common work with the Coria laboratory to post-process the primary breakup of a liquid jet using the Archer code. We underline that a similar simulation has been used in Lebas et al.(2009) to close the ELSA model.

The diameter of the injector nozzle is 100 µm, the size of the domain is 765 µm×765 µm×3060 µm and the injection velocity is 25 m/s. The characteristic of the flow is given in Table 6.5. We use a uniform 3D Cartesian grid, which is composed of $128 \times 128 \times 512$ cubic cells. Figure 6.8 illustrates the gas-liquid interface obtained at $t = 0.66$ µs. One can identify three regions of the primary breakup of the liquid jet. In the region close to the nozzle, the interface of the liquid jet has a quasi-cylindrical shape. In the middle of the domain, the interface is widely deformed and we can remark the ligaments formation. In the last part of the domain, some ligaments are detached from the liquid core and break into a large range of drop sizes. However, the liquid core is not totally atomized in this simulation. In the following, we compute the distribution of the surface curvatures in three different sub-domains. These sub-domains are illustrated by red rectangles in Figure 6.8 and they correspond to:

- Sub-domain 1 is situated close to the nozzle outlet and is given by $0 \leq z \leq 32dx$, where the z-axis is parallel to the principal jet direction.
- Sub-domain 2 is a part of domain situated in the middle of the jet and is given by $192 \leq z \leq 224dx$.
- Sub-domain 3 is situated in the downstream of the jet and is given by $480dx \leq z \leq 512dx$.



$t_a = 0.0s$ $t_b = 1.83 \times 10^{-4}s$



$t_c = 1.86 \times 10^{-4}s$ $t_d = 2.07 \times 10^{-4}s$

Figure 6.7: *Gauss and mean curvatures of the gas-liquid interface corresponding to four instants of the Plateau-Rayleigh simulation: interface curvatures in the part* $lz \geq z > lz/2$ *(red cross) and in the part* $0 \leq z \leq lz/2$ *(blue cross).*

Figure 6.8: *The gas-liquid interface of the liquid jet simulation at* $t = 0.66\,\mu\mathrm{s}$*. The rectangles show three regions considered for the extraction of curvatures.*

Figures 6.9, 6.10 and 6.11 display a color map of the SDF, as a function of the mean and Gauss curvatures, computed respectively over sub-domains 1, 2 and 3. The red line represents the spherical shapes $G = H^2$ and the dashed black line represents the cylindrical shapes $G = 0$. The curvatures are spatially averaged using the averaging method described in section 5.6 with an average scale $h = 5\Delta x$. In Figure 6.9, we have large values of the SDF for small curvatures and we can see the concentration of the contours in a very small area of the (G, H)-plan. Indeed, the curvatures in sub-domain 1 are approximately equal to the curvatures of the cylinder of diameter $D = 100\,\mu\mathrm{m}$. In Figure 6.10, we see that the distribution of the curvatures on the surface, given by the SDF in sub-domain 2, has a larger variance compared to the SDF in the region close to the nozzle outlet. First, we remark a large surface area density for small curvatures that are concentrated around the line $G = 0$, these curvatures correspond to the main liquid core. On the other hand, the large curvatures correspond to the stretching of the surface. Indeed, a non negligible part of the surface area density corresponding to larger curvatures are far away from the line $G = 0$ and the curve $G = H^2$. We mainly consider that these curvatures correspond to the ones of ligaments and large deformed drops. Finally, in Figure 6.11, we still have an important part of the surface area density concentrated around the line $G = 0$ that corresponds to the liquid core and we have further large curvatures that do not have specific region of concentration and which correspond to the curvatures of ligaments and large drops. But in this sub-domain, a third part of the surface area density associated to large curvatures is concentrated around the curve $G = H^2$ in the down part of the curve $H < 0$. These curvatures correspond mainly to some spherical and small droplets.

Hence, the SDF may provide rich information about the topological evolution of the two-phase flow together with some hints about the generated spray population. The post-processing of this jet configuration is still a work in progress, where further investigations are envisioned:

- post-processing of curvatures and their evolution for different times and in different regions of the jet.

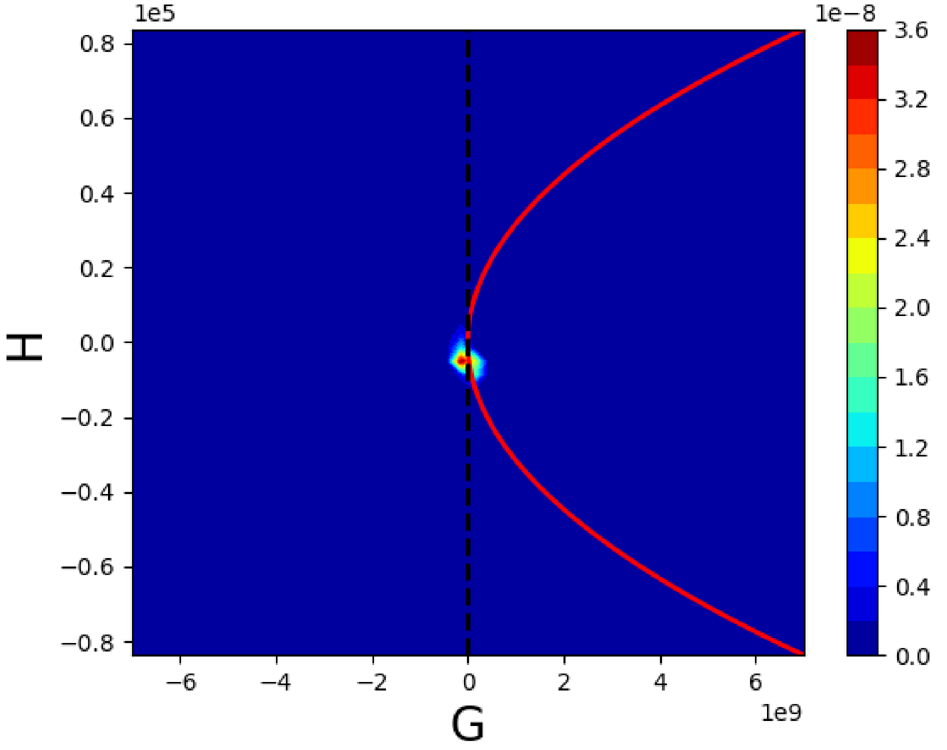


Figure 6.9: *SDF as function of* (H, G) *integrated over the sub-domain 1.*

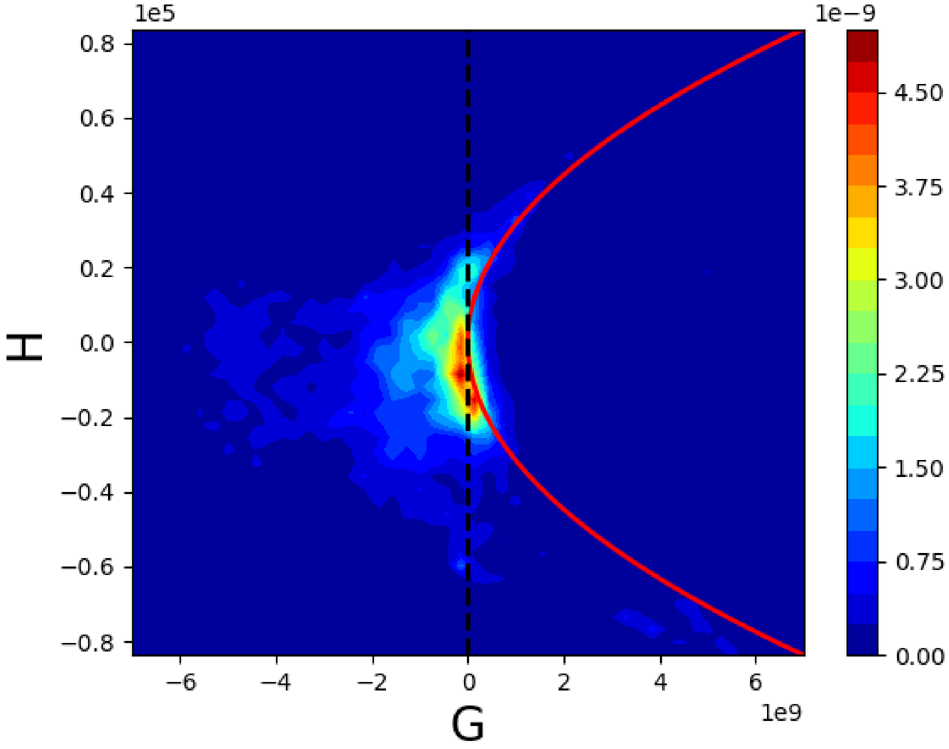


Figure 6.10: *SDF as function of* (H, G) *integrated over the sub-domain 2.*

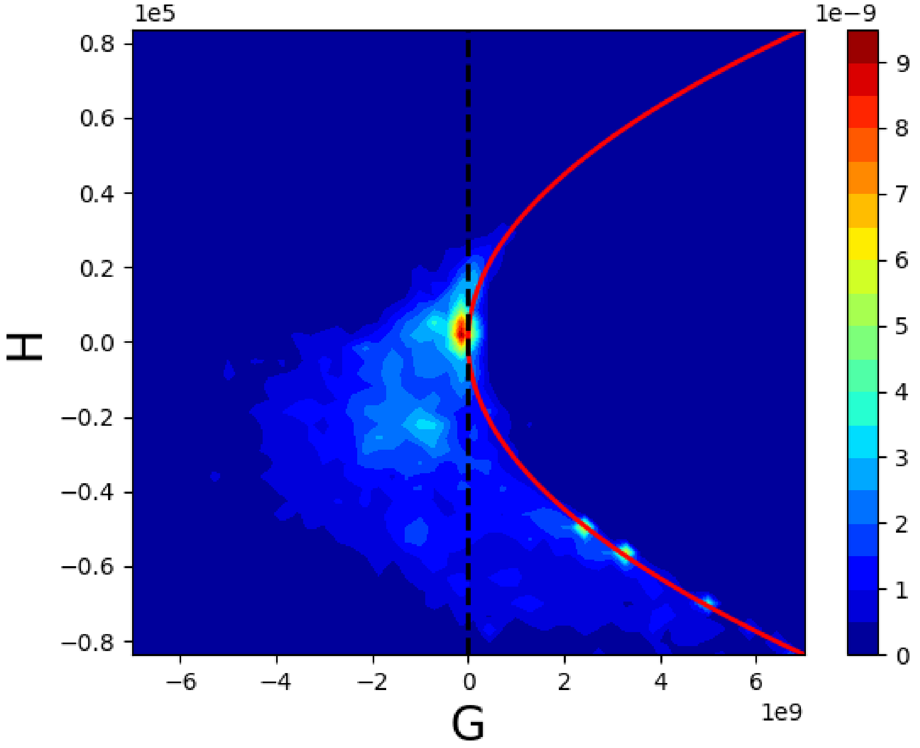


Figure 6.11: *SDF as function of* (H, G) *integrated over the sub-domain 3.*

- computing the SDF and the NDF using a time average over different iterations to obtain consistent distributions.
- defining of a convenient threshold scale δ_s for a similar configuration with respect to the reduced order model, where we use a coarsened meshes compare to DNS. Then, we would compute the exchange source terms defined in (6.13) between the resolved and unresolved scales using DNS results.

6.5 Conclusion and perspectives

Throughout this chapter, we have explained how to use the geometrical description of the interface presented in the previous chapters, in order to come up with a unified model for the disperse and separated phases. Three main points have been discussed during this presentation: the sub-scale geometrical moments, the coupling with a two-fluid model and the usefulness of DNS computations to propose closures of the model and to investigate the modeling assumptions.

Up to now, it seems that the sub-scale geometrical moments play an important role to complete the sub-scale information on the gas-liquid interface. The sub-scale variables can also enrich the two-fluid model as it was already illustrated in the work of Drui(2017). Future works Cordesse(2019); Di Battista(2020) aim at going further in this direction by including the sub-scale geometrical moments on a two-fluid model using a procedure based on Hamilton's variational principle and the second principle of thermodynamics. Finally, we underline that the different ideas proposed in this chapter are still under development and open questions require further investigations. The following lists contain some of these remaining issues:

- theoretical investigation of the threshold scale δ_s: this scale should be defined with respect to an injection benchmark simulation withe the reduced order model. It also defines the minimum mesh size needed to correctly capture the mean scales of the flow. The scales lower than δ_s should likely have some universal structures that can allow to simplify the phase space of the distribution and their evolution: $\left\langle \dot{H} \right\rangle_c$, $\left\langle \dot{G} \right\rangle_c$, $\left\langle \dot{\mathcal{S}} \right\rangle_c$ to model the primary atomization or the terms $\left\langle \dot{S} \right\rangle_c$, $\langle \dot{v} \rangle_c$, $\left\langle \dot{R} \right\rangle_c$ and $\langle \dot{\chi}/\chi \rangle_c$, to model the secondary breakup.
- influence of the averaged scale h on the DNS curvature post-processing and on the sub-scales modeling: as it was illustrated in the Plateau-Rayleigh simulation, this scale has an influence on the post-processing results. Without averaging, the curvatures contain a large noise, which makes its difficult to analyze. On the contrary, a large spatial averaging leads to some important loss in structures of these curvatures. In terms of modeling, we would like to have enough information on the interface in the separated phases to capture the atomization, while in disperse phase we estimate that the averaged internal variables over droplet surface are sufficient to model the dynamics of the droplets from a macroscopic point of view.
- evaluating the exchange fluxes between a two-fluid model and the sub-scales equations: the source terms expressed in (6.13) can be first computed directly by using DNS results. The results obtained from such a DNS post-processing will be used to relate these source terms to the variables of the used two-fluid model.

Finally, let us underline that it is the first time DNS data are analyzed with curvatures evaluation. This post-processing of the gas-liquid interface is consistent with geometrical and topological invariants (see chapter 5). This should provide a tool to better understand and analyze the physics of such flows.

Part III

Numerical methods

Chapter 7

Numerical resolution of the transport equations of the fractional moments model

7.1 Introduction

In this chapter, we focus on the numerical resolution of the transport part of the fractional moments model presented in chapter 4. In fact, we can split the resolution of the transport part (LHS of system (4.24)) from the source terms (RHS of system (4.24)), by mean of operator splitting techniques Doisneau et al.(2014); Descombes et al.(2014). The numerical resolution of the source terms is treated in chapter 8. The transport part of the system of equations (4.24) is similar to the one of the EMSM model Kah et al.(2012). This system of equations can be written as follows:

$$\partial_t \boldsymbol{U} + \nabla_{\boldsymbol{x}} \cdot (\boldsymbol{\mathcal{F}}(\boldsymbol{U})) = \boldsymbol{0}, \tag{7.1}$$

where $\boldsymbol{U}^t = (\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, \mathrm{m}_{2/2}, \mathrm{m}_{3/2}, \mathrm{m}_{2/2}\boldsymbol{u}^t)$, $\boldsymbol{\mathcal{F}}(\boldsymbol{U}) = (\boldsymbol{\mathcal{F}}_1(\boldsymbol{U}) \dots \boldsymbol{\mathcal{F}}_d(\boldsymbol{U}))$, d is the space dimension and $\boldsymbol{\mathcal{F}}_\alpha(\boldsymbol{U}) = u_\alpha \boldsymbol{U}$ for $1 \leq \alpha < d$.

The system of equations (7.1) has similar mathematical properties as the pressureless gas dynamic (PGD) model of de Chaisemartin et al.(2009). Indeed, the system is only weakly hyperbolic: it has only one real eigenvalue and four linearly independent eigenvectors in each direction, instead of $4+d$ ($4+d$ is the equations number). In the case of Particle Trajectory Crossing, the model implies the segregation of the droplets in infinitely fine zones, which is known as δ-shocks. For the numerical resolution of this system, we need to cope with different challenges. First, accurate resolution is required to limit the numerical diffusion. To satisfy this first requirement at a reasonable computational cost, high order numerical schemes can be used. The second issue concerns the robustness of the numerical scheme. High order numerical schemes can cause oscillations of the solution near the zones where the solution is singular (δ-shocks). A familiar solution to this issue consists in using artificial diffusion to suppress the spurious oscillations. However, this type of methods depends on some parameters and shock capturing sensors, which need to be adapted for each situation. Also the accuracy of the method is generally reduced. It is then important to develop robust numerical schemes, which do not depend on additional parameters and preserve the order accuracy of regular solutions. Finally, the realizability of the moments (a non negative NDF can be associated to every set of moments) is an essential condition to ensure a physical solution and to allow everywhere the well-posedness of the ME algorithm presented in chapter 4. A naive numerical schemes can update the moments set in a wrong way that leads to a non-realizable set of moments.

In the present work, we propose two numerical scheme classes. The first one is a finite volume discretization based on a kinetic approach Bouchut et al.(2003); de Chaisemartin et al.(2009). We call this class of schemes the Kinetic Finite Volume (KFV) class. In this method, we compute the numerical flux based on the exact solution of a kinetic transport equation. To achieve high order accuracy, non-constant reconstruction is needed to evaluate the numerical fluxes. However, the reconstruction should preserve the moments vector within the moment space to ensure the realizability of the numerical scheme. This condition is satisfied by using a slope limitation on the reconstructed canonical moments (see chapter 4 for the definition of the canonical moments), in the same way as it was proposed in the first time by Kah et al.(2012). To step forward in improving the accuracy of the numerical resolution and developing higher order schemes, we use a Rung-Kutta Discontinuous Galerkin discretization Cockburn and Shu(1998), as the second class of the schemes used to solve system (7.1). Indeed, Discontinuous Galerkin (DG) scheme is a promising alternative compared to standard finite volume methods. This choice can be justified for two reasons. First, finite volume methods require large extended stencils to achieve high order accuracy, which can be restrictive for the scalability of parallel computations. On the contrary, DG methods use a compact stencil since the degrees of freedom are store within each computational cell (similar idea as the Finite Element method). Finally, to ensure a realizable resolution of the moments, we adopt a similar limitation method as the one proposed in Zhang and Shu(2011); Zhang et al.(2012).

7.2 Realizable Kinetic Finite Volume (KFV) schemes

The use of Cartesian grids in physical space enables to use a dimensional splitting algorithm. Dimensional splitting consists in solving the system of equations separately and successively in each direction during a time step Δt. In this case, we can reuse a 1D finite volume scheme to update the solution. In the following, we choose to present the scheme in a two dimensional space in order to lighten the notations and we denote by $\boldsymbol{u} = (u, v)$ the 2D velocity vector.

Let $\square_\gamma^{\Delta t}\bullet$ be an operator associated to a 1D finite volume scheme, which can be used to update the solution in a direction $\gamma = x, y$ during a time step Δt. The dimensional splitting leads to an additional error in the numerical resolution. The order of the additional error depends on the splitting algorithm. In the present work, we use two types of splitting:

- Godunov or Lie splitting method is a first order method, where the solution is updated successively for each direction as follows:

$$\boldsymbol{U}_i^{n,*} = \square_x^{\Delta t}\boldsymbol{U}_i^n, \quad \boldsymbol{U}_i^{n+1} = \square_y^{\Delta t}\boldsymbol{U}_i^{n,*}. \tag{7.2}$$

- Strang splitting method is a second order accurate method when the finite volume 1D operator $\square_\gamma^{\Delta t}\bullet$ is also of second order. The solution is updated in this case as follows:

$$\boldsymbol{U}_i^{n+1} = \square_x^{\Delta t/2}\square_y^{\Delta t}\square_x^{\Delta t/2}\boldsymbol{U}_i^n. \tag{7.3}$$

In the following, we detail the finite volume 1D operator $\square_x^{\Delta t}\bullet$, which corresponds to a finite volume discretization of the free transport equations in one direction (we focus on the x-direction here):

$$\begin{array}{rcl} \partial_t \mathrm{m}_{k/2} + \partial_x(\mathrm{m}_{k/2} u) & = & 0, \\ \partial_t(\mathrm{m}_{2/2}\boldsymbol{u}) + \partial_x(\mathrm{m}_{2/2} u \boldsymbol{u}) & = & 0, \end{array} \tag{7.4}$$

where $k = 0, \ldots, 3$.

As it was discussed in the introduction, we encounter two main difficulties in the resolution of this system. The first one corresponds to the possible appearance of strong singularities (known as δ-shocks), which occur when the monokinetic assumption is violated. This can happen when trajectory crossings take place and lead to particles accumulation in a very small volume. Then, the numerical scheme has to be able to capture these shocks and ensure a stable solution. The second issue concerns the realizability of the numerical scheme (the moment set approximation can be associated to a non-negative NDF). In the following, we show how to cope with these issues using a kinetic approach.

7.2.1 Derivation of 1D finite volume kinetic scheme for fractional moments equations

We consider a uniform discretization $(x_i)_{i=1,\ldots N_x}$ of a 1D compact domain $[x_{min}, x_{max}]$. Such that we have:

$$x_{min} + \Delta x/2 = x_1 < x_2 < \ldots < x_{N_x} = x_{max} - \Delta x/2, \quad \text{and} \quad \frac{x_{max} - x_{min}}{N_x} = \Delta x, \tag{7.5}$$

where x_i is the center of cell $i \in [1, N_x]$. We denote by $x_{i+1/2} = x_i + \Delta x/2$ (respectively $x_{i-1/2} = x_i - \Delta x/2$) the right (respectively the left) face of cell i. In the finite volume representations, we look for an approximation of the solution at a discrete time t_n:

$$\boldsymbol{U}_i^n \simeq \frac{1}{\Delta x} \int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{U}(t_n, x) dx, \tag{7.6}$$

where $t_n = \sum_{j=1}^n \Delta t_j$ (with $t_0 = 0$) and $\Delta t_j = t_j - t_{j-1}$ is the time step. The time step should satisfy a Courant-Friedrichs-Lewy (CFL) condition to ensure the stability of the numerical solution. By integrating equation (7.4) over time $[t_n, t_{n+1}]$ and space $[x_{i-1/2}, x_{i+1/2}]$, the finite volume numerical scheme reads:

$$\boldsymbol{U}_i^{n+1} = \boldsymbol{U}_i^n - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+1/2} - \boldsymbol{F}_{i-1/2}\right). \tag{7.7}$$

The numerical fluxes $\boldsymbol{F}_{i\pm1/2}$ evaluated at the faces are an approximation of the exact mean-time flux:

$$\boldsymbol{F}_{i\pm1/2} \simeq \frac{1}{\Delta t} \int_{t_n}^{t_{n+1}} \boldsymbol{\mathcal{F}}(t, x_{i\pm1/2}) dt \tag{7.8}$$

The numerical flux $\boldsymbol{F}_{i\pm1/2}$ can be approximated using the left and the right states with respect to the considered face:

$$\begin{array}{rcl} \boldsymbol{F}_{i+1/2} & = & \boldsymbol{F}(\boldsymbol{U}_i, \boldsymbol{U}_{i+1}) \\ \boldsymbol{F}_{i-1/2} & = & \boldsymbol{F}(\boldsymbol{U}_{i-1}, \boldsymbol{U}_i) \end{array} \tag{7.9}$$

de Chaisemartin(2009) proposed a finite volume kinetic scheme to solve the monokinetic multi-fluid model based on the same approach as in Bouchut et al.(2003), used to solve the PGD system of equations. Following the same idea, Kah et al.(2012) proposed an extention of the kinetic numerical scheme for higher order size-moments model. In the following, we use this approach to express the numerical fluxes of the present system of equations (7.1). The main steps to derive the kinetic scheme for system (7.4) are briefly presented as follows:

A. We consider the equivalent kinetic system to the pressureless system (7.4), as was proposed in Bouchut et al.(2003):

$$\begin{cases} \partial_t f + \partial_x (c_x f) = 0 \text{ , and} \\ f(t,x,\boldsymbol{c},S) = n(t,x,S)\delta(\boldsymbol{c}-\boldsymbol{u}), \end{cases} \tag{7.10}$$

B. We express the exact finite volume fluxes as a function of the NDF $f(t,x,\boldsymbol{c},S)$.

$$\frac{1}{\Delta t}\int_{t_n}^{t_{n+1}} \mathcal{F}(t,x_{i\pm1/2})dt = \frac{1}{\Delta t}\int_{t_n}^{t_{n+1}}\int_{S=0}^{1}\int_{\boldsymbol{c}=(c_x,c_y)} c_x \boldsymbol{V}(S,\boldsymbol{c}) f(t,x,\boldsymbol{c},S)dSd\boldsymbol{c}, \tag{7.11}$$

where $\boldsymbol{V}(S,\boldsymbol{c}) = (1, S^{1/2}, S, S^{3/2}, Sc_x, Sc_y)$.

C. We split the fluxes in two integral parts: the first $\boldsymbol{F}^{+}_{i+1/2}$ (resp the second $\boldsymbol{F}^{-}_{i+1/2}$) corresponds to the droplet of positive velocity $c_x \geq 0$ (resp negative velocity $c_x < 0$). Then, we use the exact solution of the kinetic system (7.10), to express the fluxes as a function of the NDF at time t_n:

$$f(t,x,\boldsymbol{c},S) = f(t_n, x - c_x(t-t_n), \boldsymbol{c}, S) \tag{7.12}$$

for $t \in [t_n, t_{n+1}]$.

D. Finally, the fluxes are expressed as functions of the known moments and velocities at $t = t_n$:

$$\begin{aligned} \boldsymbol{F}^{+}_{i+1/2} &= \frac{1}{\Delta t}\int\limits_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{U}(t_n,x)\mathbb{1}_{\Sigma^+}(x)dx, \\ \boldsymbol{F}^{-}_{i+1/2} &= \frac{1}{\Delta t}\int\limits_{x_{i+1/2}}^{x_{i+3/2}} \boldsymbol{U}(t_n,x)\mathbb{1}_{\Sigma^-}(x)dx, \end{aligned} \tag{7.13}$$

where $\Sigma^{\pm} = \{x', \pm(x_{i+1/2} - \Delta t u(t_n,x')) < \pm x'\}$. The global numerical flux is then:

$$\boldsymbol{F}_{i\pm1/2} = \boldsymbol{F}^{+}_{i\pm1/2} + \boldsymbol{F}^{-}_{i\pm1/2}. \tag{7.14}$$

7.2.2 First order scheme

For a first order scheme, we consider a constant piecewise reconstruction for the moments and for the velocity (see Figure 7.1-left). In this case, the numerical scheme turns to be a simple upwind scheme, where the numerical fluxes express as follows:

$$\boldsymbol{F}_{i+1/2} = \begin{pmatrix} m^n_{0,i} \\ m^n_{1/2,i} \\ m^n_{1,i} \\ m^n_{3/2,i} \\ m^n_{1,i}u^n_i \\ m^n_{1,i}v^n_i \end{pmatrix} max(u^n_i, 0) + \begin{pmatrix} m^n_{0,i+1} \\ m^n_{1/2,i+1} \\ m^n_{1,i+1} \\ m^n_{3/2,i+1} \\ m^n_{1,i+1}u^n_{i+1} \\ m^n_{1,i}v^n_{i+1} \end{pmatrix} min(u^n_{i+1}, 0), \tag{7.15}$$

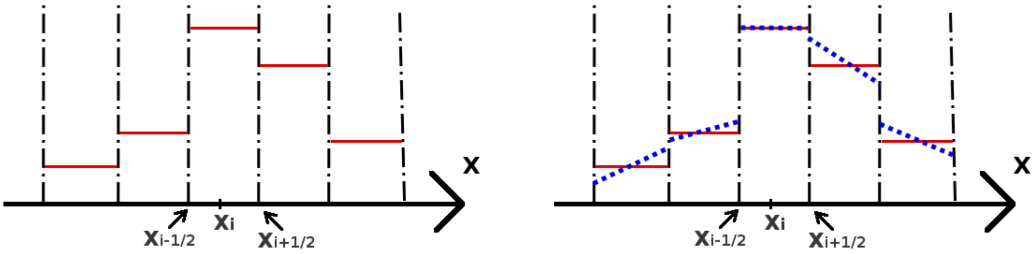


Figure 7.1: *Sketch of finite volume representations in a cell. Left piecewise-constant representation used at first order. Right piecewise-linear reconstruction of the solution to achieve second order accuracy.*

The numerical scheme is stable under a CFL condition which reads:

$$\frac{\Delta t}{\Delta x} = CFL \max |u_i^n|, \tag{7.16}$$

where $CFL \leq 1$. Under this condition, we can also show that the updated solution at t_{n+1} (7.7) in the cell i can be written as a convex combination of the solution at t_n in the cells $i-1, i,$ and $i+1$. Since the moments space is a convex space, the numerical scheme is intrinsically realizable.

7.2.3 Second order scheme

Usually, high order finite volume schemes use non-constant reconstructions of the transported variables (see figure 7.1-right) to better evaluate the fluxes at the interfaces of the cells. The main difficulty consists in reconstructing a realizable moments solution. Kah et al.(2012) designed a realizable second order kinetic scheme for the EMSM model. Instead of reconstructing directly the integer moments, the authors proposed to use a linear reconstruction of the canonical moments associated to integer moments. Indeed, the canonical moments live in the simple space $[0,1]^N$ ($N = 3$ for the EMSM model), while the moment space has a rather complex geometry. Furthermore, the canonical moments are also simply transported variables:

$$\partial_t p_k + u \partial_x p_k = 0, \tag{7.17}$$

where p_k are the canonical integer moments defined in (4.6). Therefore, these quantities satisfy a maximum principle. In the following, we adopt the same approach with some slight adaptations to the fractional moments case.

7.2.3.1 Reconstruction of the transported variables

The reconstructed variables are the moment $\mathrm{m}_{0/2}$, the canonical moments associated this time to the fractional moments $(p_{1/2}, p_{2/2}, p_{3/2})$ (see the definition in (4.28)) and the two velocity components (u, v).

$$\left\{\begin{array}{rcl} \mathrm{m}_{0/2}(x) & = & \overline{\mathrm{m}_{0/2,i}} + Dm_{0,i}(x - x_i), \\ p_{1/2}(x) & = & \overline{p_{1/2,i}} + Dp_{1,i}(x - x_i), \\ p_{2/2}(x) & = & \overline{p_{2/2,i}} + Dp_{2,i}(x - x_i), \\ p_{3/2}(x) & = & \overline{p_{3/2,i}} + Dp_{3,i}(x - x_i), \\ u(x) & = & \overline{u_i} + Du_i(x - x_i), \\ v(x) & = & \overline{v_i} + Dv_i(x - x_i), \end{array}\right. \tag{7.18}$$

where $x \in [x_{i-1/2}, x_{i+1/2}]$. Generally the quantities with the bar are different from the cell quantities $p_{k/2,i}$, u_i and v_i and they are determined depending on the slopes and the following conservation properties:

$$\begin{array}{rcl} \mathrm{m}^n_{1/2,i} & = & \dfrac{1}{\Delta x}\displaystyle\int_{x_{i-1/2}}^{x_{i+1/2}} \mathrm{m}_{0/2}(x)p_{1/2}(x)dx, \\ \mathrm{m}^n_{2/2,i} & = & \dfrac{1}{\Delta x}\displaystyle\int_{x_{i-1/2}}^{x_{i+1/2}} \mathrm{m}_{0/2}(x)p_{1/2}(x)[(1-p_{1/2})p_{2/2}+p_{1/2}](x)dx, \\ \mathrm{m}^n_{3/2,i} & = & \dfrac{1}{\Delta x}\displaystyle\int_{x_{i-1/2}}^{x_{i+1/2}} \mathrm{m}_{0/2}p_{1/2}\left\{(1-p_{1/2})(1-p_{2/2})p_{2/2}p_{3/2} + [(1-p_{1/2})p_{2/2}+p_{1/2}]^2\right\}(x)dx, \\ m^n_{2/2,i}u^n_i & = & \dfrac{1}{\Delta x}\displaystyle\int_{x_{i-1/2}}^{x_{i+1/2}} \mathrm{m}_{0/2}(x)p_{1/2}(x)[(1-p_{1/2})p_{2/2}+p_{1/2}](x)u(x)dx. \end{array} \tag{7.19}$$

Compared to the expressions developed in the case of the EMSM model, only the last integral expression is different. In fact, the velocity is weighted by the moment $\mathrm{m}_1 = \mathrm{m}_{2/2}$ for both models. But with fractional moments, $\mathrm{m}_{2/2}$ acts as a second order moment. For this reason, the expression of the moment $\mathrm{m}_{2/2}$ as a function of the canonical moments is different from the one in the case of the integer moments.

Following the same derivation as in Kah et al.(2012), we show that the bar terms can be written as follows:

$$\begin{array}{rcl} \overline{p_{k/2,i}} & = & a_{k,i} + b_{k,i}Dp_{k,i}, \\ \overline{u_i} & = & a_{u,i} + b_{u,i}Du_i, \end{array} \tag{7.20}$$

where for each k, $a_{k,i}$ and $b_{k,i}$ are independent of $Dp_{k,i}$, and $a_{u,i}$ and $b_{u,i}$ are independent of Du_i.

7.2.3.2 Slope limitation

In order to satisfy the maximum principle for the transported quantities (the canonical moments and the velocity) and the positivity of the number density $\mathrm{m}_{0/2}$, the slopes should be calculated carefully. Following the development done in Kah et al.(2012), the slopes are calculated as follows:

$$\begin{array}{rl} Dm_{0,i} & = \phi(\mathrm{m}^n_{0/2,i-1}, \mathrm{m}^n_{0/2,i}, \mathrm{m}^n_{0/2,i+1}) \min\left(\dfrac{|\mathrm{m}^n_{0/2,i+1} - \mathrm{m}^n_{0/2,i}|}{\Delta x}, \dfrac{|\mathrm{m}^n_{0/2,i} - \mathrm{m}^n_{0/2,i-1}|}{\Delta x}, \dfrac{2\mathrm{m}^n_{0/2,i}}{\Delta x}\right), \\ Dp_{k,i} & = \phi(p^n_{k/2,i-1}, p^n_{k/2,i}, p^n_{k/2,i+1}) \min\left(\dfrac{|p^n_{k/2,i+1} - a_{k,i}|}{\Delta x + 2b_{k,i}}, \dfrac{|a_{k,i} - p^n_{k/2,i-1}|}{\Delta x - 2b_{k,i}}\right), \\ Du_i & = \phi(u^n_{i-1}, u^n_i, u^n_{i+1}) \min\left(\dfrac{|u^n_{i+1} - u^n_i|}{\Delta x + 2b_{u,i}}, \dfrac{|u^n_i - u^n_{i-1}|}{\Delta x - 2b_{u,i}}, \dfrac{1}{dt}\right), \end{array}$$

(7.21)

where $\phi(a,b,c) = 1/2(sgn(b-a) + sgn(c-b))$ and $sgn(x)$ is the sign of x.

Using equations (7.19), (7.20) and (7.21), the slopes and the bar variables can be expressed as functions of the current and neighbor cell variables. However, these algebra relations are quite heavy. Therefore, their calculation is achieved using the Maple software.

7.2.3.3 Fluxes Computation

After computing the slopes and the bar variables, the fluxes can be computed as follows:

$$\boldsymbol{F}^{+}_{i+1/2} = \frac{1}{dt}\int_{x^L_{i+1/2}}^{x_{i+1/2}} \mathrm{m}_{0/2} \begin{pmatrix} 1 \\ p_{1/2} \\ p_{1/2}[(1-p_{1/2})p_{2/2} + p_{1/2}] \\ p_{1/2}\left\{(1-p_{1/2})(1-p_{2/2})p_{2/2}p_{3/2} + [(1-p_{1/2})p_{2/2} + p_{1/2}]^2\right\} \\ p_{1/2}[(1-p_{1/2})p_{2/2} + p_{1/2}]u \\ p_{1/2}[(1-p_{1/2})p_{2/2} + p_{1/2}]v \end{pmatrix} dx, \tag{7.22}$$

and

$$\boldsymbol{F}^{-}_{i+1/2} = -\frac{1}{dt}\int_{x_{i+1/2}}^{x^R_{i+1/2}} \mathrm{m}_{0/2} \begin{pmatrix} 1 \\ p_{1/2} \\ p_{1/2}[(1-p_{1/2})p_{2/2} + p_{1/2}] \\ p_{1/2}\left\{(1-p_{1/2})(1-p_{2/2})p_{2/2}p_{3/2} + [(1-p_{1/2})p_{2/2} + p_{1/2}]^2\right\} \\ p_{1/2}[(1-p_{1/2})p_{2/2} + p_{1/2}]u \\ p_{1/2}[(1-p_{1/2})p_{2/2} + p_{1/2}]v \end{pmatrix} dx, \tag{7.23}$$

such that

$$\begin{aligned} x^L_{i+1/2} &= x_{i+1/2} - dt\frac{(\bar{u}_i + \frac{\Delta x}{2}Du_i)^+}{1 + dtDu_i}, \\ x^R_{i+1/2} &= x_{i+1/2} - dt\frac{(\bar{u}_{i+1} - \frac{\Delta x}{2}Du_{i+1})^-}{1 + dtDu_{i+1}}, \end{aligned} \tag{7.24}$$

where $(\psi)^+ = 1/2(|\psi| + \psi)$ and $(\psi)^- = 1/2(-|\psi| + \psi)$.

The expressions inside the integrals are polynomial functions of x of order up to 6, their calculation can be achieved by using four points of the Gauss-Legendre quadrature.

The CFL condition for the second order scheme reads:

$$\frac{\Delta t}{\Delta x} = CFL \max\left(|\overline{u_i}| + |Du_i|\Delta x/2\right) \tag{7.25}$$

where $CFL \leq 1$ is defined by the user.

7.3 Realizable discontinuous Galerkin method

The original DG method was introduced by Reed and Hill Reed and Hill(1973). Recently the DG method has been widely used in different numerical simulation fields and especially in computational fluid dynamics to achieve high order accuracy. Zhang & Shu, in a series of papers Zhang and Shu(2010); Zhang and Shu(2011); Zhang et al.(2012), proposed to use the Runge Kutta Discontinuous Galerkin (RKDG) method for the numerical resolution of hyperbolic equations, where the Runge Kutta method is used for the time integration. The authors were able to combine high order accuracy and preserve certain convex constraints of the solution by using a positivity limiting method. They have used this method to ensure the positivity of the mass density and of the pressure in the Euler equation. The extension of this framework to weakly hyperbolic equations (PGD system) was initiated in Larat et al.(2012). The same idea can be extended to ensure the realizability of the numerical resolution of the fractional size-moments. In the following, we present the general DG discretization of system (7.1) in 2D. Then, we show how to use the limitation method of Zhang & Shu to ensure the realizability of the fractional moments . Finally, the same idea will be used to satisfy the maximum principle on the velocity, in order to obtain robust schemes.

7.3.1 General DG discretization

In the following, we consider system (7.1) in 2D. We partition the domain by non-overlapping rectangular cells.

$$\Omega = \bigcup_{m=1}^{N_{cell}} \mathcal{K}_m, \quad (\forall m \neq m') \;\; dim(\mathcal{K}_m \cap \mathcal{K}_{m'}) < 2 \tag{7.26}$$

where N_{cell} is the number of cells and $\mathcal{K}_m = [x_{m-1/2}, x_{m+1/2}] \times [y_{m-1/2}, y_{m+1/2}]$ is the rectangular domain of cell $m \in \{1, \ldots, N_{cell}\}$. For the sake of simplicity, we suppose a Cartesian uniform mesh: $\Delta x = x_{m+1/2} - x_{m-1/2} = y_{m+1/2} - y_{m-1/2}$. For a DG discretization of order p, we look to approximate the solution of system (7.1) by a piecewise polynomial $\boldsymbol{U}_h(t, .)$ of degree $k = p - 1$, where the restriction of the solution to each cell m can be written as a polynomial function of degree $k = p - 1$ in each space direction[1]. Let $(\phi_i)_{1 \leq i \leq (k+1)^2}$ be a basis of polynomial functions of degree k in each direction of the space defined in $[-1/2, 1/2]^2$. We write the numerical solution within cell m as follows:

$$\boldsymbol{U}_h(t, \boldsymbol{x}) = \sum_{i=1}^{(k+1)^2} \boldsymbol{W}_i^m(t) \phi_i(\Xi_m(\boldsymbol{x})), \quad \boldsymbol{x} \in \mathcal{K}_m \tag{7.27}$$

where the coefficients $\boldsymbol{W}_i^m(t)$ are called the degrees of freedom in cell m, $\Xi_m(\boldsymbol{x}) = \dfrac{\boldsymbol{x} - \boldsymbol{x}_m}{\Delta x}$ is the one-to-one transformation between the cell domain $\mathcal{K}_m$ and the canonical cell $[-1/2, 1/2]^2$ and $\boldsymbol{x}_m$ is the center of cell m. According to DG formulation, the numerical solution is the unique solution of the weak formulation of system (7.1), which can be simplified in the following form:

$$\begin{aligned} \sum_{j=1}^{(k+1)^2} \Delta x^2 \boldsymbol{\mathcal{M}}_{i,j} \frac{d\boldsymbol{W}_j^m}{dt} \;\; = \;\; & \Delta x \int\limits_{\xi \in [-1/2,1/2]^2} \boldsymbol{\mathcal{F}}(\boldsymbol{U}_h(t, \Xi_m^{-1}(\boldsymbol{\xi})) \cdot \nabla_{\xi}(\phi_i(\boldsymbol{\xi}))d\boldsymbol{\xi} \\ & - \Delta x \sum_{e \in D} \int\limits_{\xi=-1/2}^{1/2} \boldsymbol{\mathcal{F}}(\boldsymbol{U}_h(t, \widetilde{\boldsymbol{x}}_{m,e}(\xi)) \cdot \boldsymbol{n}_{m,e} \phi_i(\Xi_m(\widetilde{\boldsymbol{x}}_{m,e}(\xi)))d\xi, \end{aligned} \tag{7.28}$$

---

[1] Polynomials function of degree k in each direction in a multi-space dimension (in 2D $p(x, y) = \sum_{i,j=0}^{k} a_{i,j} x^i y^j$) are different than polynomial of degree k in all directions (in 2D $p(x, y) = \sum_{0 \leq i+j \leq k} a_{i,j} x^i y^j$). In DG method, we often use the second option, which is the optimal one to achieve an order of accuracy $p = k + 1$ with a minimum degrees of freedom. We made this non-optimal choice to simplify the implementation. In future work, it will be more convenient to consider the second one.

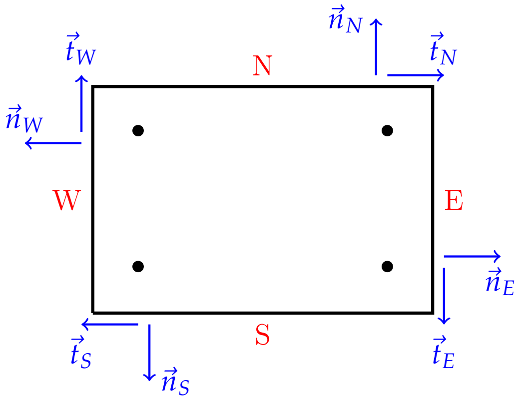


Figure 7.2: *Illustration of the cell edges and of the orientation of their normal and tangential vectors used in DG discretization scheme (7.28). The points in the interior represents the Gauss-Legendre quadrature points for $k = 2$.*

where $\mathcal{M}_{i,j} = \int_{\xi\in[-1/2,1/2]^2} \phi_i\phi_j d\boldsymbol{\xi}$ is the mass matrix. This matrix is independent of cell m. And $D = \{N, W, S, E\}$ is the set of cell edges as illustrated in Figure 7.2, $\boldsymbol{n}_{m,e}$ is the normal vector of edge e oriented outward of cell m and $\widetilde{\boldsymbol{x}}_{m,e}(\xi)$ are the positions along of edge $e \in D$ for $\xi \in [-1/2, 1/2]$, which are defined as follows:

$$\widetilde{\boldsymbol{x}}_{m,e}(\xi) = \boldsymbol{x}_m + \Delta x/2\boldsymbol{n}_{m,e} + (\xi\Delta x)\boldsymbol{t}_{m,e}, \tag{7.29}$$

where $\boldsymbol{t}_{m,e}$ is a tangent vector of edge e (see Figure 7.2).

The solution is discontinuous at the edges, what allows DG methods to capture shocks of hyperbolic equations. The flux $\mathcal{F}(\boldsymbol{U}_h(t, \widetilde{\boldsymbol{x}}_e(\xi))$ at edge e is approximated with a numerical flux, as for finite volume schemes. The numerical flux, computed at a given edge e, depends on the following left and right state of the solution with respect to the edge e:

$$\begin{aligned} \boldsymbol{U}_h^-(t, \widetilde{\boldsymbol{x}}_{m,e}(\xi)) &= \lim_{\gamma\to 0^-} \boldsymbol{U}_h(t, \widetilde{\boldsymbol{x}}_e(\xi) + \gamma\boldsymbol{n}_{m,e}), \\ \boldsymbol{U}_h^+(t, \widetilde{\boldsymbol{x}}_{m,e}(\xi)) &= \lim_{\gamma\to 0^+} \boldsymbol{U}_h(t, \widetilde{\boldsymbol{x}}_e(\xi) + \gamma\boldsymbol{n}_{m,e}). \end{aligned} \tag{7.30}$$

In the following, we use the upwind numerical flux such that we have:

$$\mathcal{F}(\boldsymbol{U}_h(t, \widetilde{\boldsymbol{x}}_e(\xi)) \cdot (\pm\boldsymbol{e}_\gamma) = \pm\boldsymbol{F}_\gamma(\boldsymbol{U}_h^-(t, \widetilde{\boldsymbol{x}}_{m,e}(\xi)), \boldsymbol{U}_h^+(t, \widetilde{\boldsymbol{x}}_{m,e}(\xi))), \tag{7.31}$$

where $\gamma = x, y$ are the two space directions and the numerical flux $\boldsymbol{F}_\gamma(.,.)$ is defined in (7.15).

7.3.2 Polynomial basis choice and quadrature integration

In order to simplify the presentation of the method in 2D, we use two indexes for the degrees of freedom $(\boldsymbol{W}_{\alpha,\beta})_{1\le\alpha,\beta\le k+1}$ and the polynomial basis $(\phi_{\alpha,\beta})_{1\le\alpha,\beta\le k+1}$. We also consider that each cell is indexed by two variables $m \in [\![1; N_{cell}]\!] \to (m_x, m_y) \in [\![1; N_x]\!] \times [\![1; N_y]\!]$ one for each direction:

$$\begin{aligned} m_y &= \left[\frac{m}{N_x}\right] \\ m_x &= m - N_x m_y \end{aligned} \tag{7.32}$$

where N_x (resp N_y) is the number of cells in direction of x (resp y) and $N_{cell} = N_x N_y$.

For the polynomial basis, we choose the Lagrangian polynomial basis associated to 2D abscissa points of Gauss Legendre quadrature in $[-1/2, 1/2]^2$:

$$\phi_{\alpha,\beta}(\boldsymbol{\xi}) = (\xi_x, \xi_y)) = L_\alpha(\xi_x) L_\beta(\xi_y) \tag{7.33}$$

and

$$L_\alpha(\xi) = \frac{\prod_{\beta \neq \alpha}(\xi - \xi_\beta)}{\prod_{\beta \neq \alpha}(\xi_\alpha - \xi_\beta)}, \tag{7.34}$$

where $\boldsymbol{\xi}_{\alpha',\beta'} = (\xi_{\alpha'}, \xi_{\beta'})$ and $(\xi_{\alpha'})_{1 \le \alpha' \le k+1}$ is the abscissas of the $(k+1)$-point Gauss-Legendre quadrature in $[-1/2, 1/2]$. In this case, the degrees of freedom are equal to the values of the solution at the points $\boldsymbol{x}_{\alpha,\beta} = \boldsymbol{\Xi}_m(\boldsymbol{\xi}_{\alpha,\beta})$ for each cell m:

$$U_h(t, \boldsymbol{x}_{\alpha,\beta}) = \boldsymbol{W}^m_{\alpha,\beta}. \tag{7.35}$$

This choice will simplify the resolution of the source terms (see appendix C), where we solve directly an Ordinary Differential equation (ODE) for each degrees of freedom separately. Solver of ODE source terms is presented in chapter 8.

The integrals in (7.28) can be approximated by quadratures of sufficient accuracy. An analysis of the requirement on the quadrature choice, presented in Cockburn et al.(1990), shows that a quadrature that is exact for polynomials of order $2k+1$, is sufficient to preserve the order accuracy of the DG scheme. In this work, we use $k+1$-point Gauss-Legendre quadrature in each direction to evaluate respectively 2D and 1D integrals. This integral approximation is an exact evaluation of the integral for the polynomial functions of degree $2k+1$. The numerical scheme (7.28) becomes:

$$\begin{aligned}
\left(\boldsymbol{\mathcal{M}}\frac{d\boldsymbol{W}^m}{dt}\right)_{\alpha,\beta} \quad = \quad & \frac{1}{\Delta x}\sum_{0 \le \alpha', \beta' \le k+1} w_{\alpha'} w_{\beta'} \left(\boldsymbol{\mathcal{F}}(\boldsymbol{W}_{\alpha',\beta'}) \cdot \boldsymbol{\nabla}_\xi \left(\phi_{\alpha,\beta}(\boldsymbol{\xi}_{\alpha',\beta'})\right)\right) \\
& - \frac{1}{\Delta x}\sum_{0 \le \beta' \le k+1} w_{\beta'} \left(\phi_{\alpha,\beta}(\boldsymbol{\xi}_{m_x+1/2,\beta'})(\boldsymbol{F}_x(\boldsymbol{W}^L_{m_x+1/2,\beta'}, \boldsymbol{W}^R_{m_x+1/2,\beta}) - \right. \\
& \left. \phi_{\alpha,\beta}(\boldsymbol{\xi}_{m_x-1/2,\beta'})\boldsymbol{F}_x(\boldsymbol{W}^L_{m_x-1/2,\beta'}, \boldsymbol{W}^R_{m_x-1/2,\beta'})\right) \\
& - \frac{1}{\Delta x}\sum_{0 \le \alpha' \le k+1} w_{\alpha'} \left(\phi_{\alpha,\beta}(\boldsymbol{\xi}_{\alpha',m_y+1/2})\boldsymbol{F}_y(\boldsymbol{W}^L_{\alpha',m_y+1/2}, \boldsymbol{W}^R_{\alpha',m_y+1/2}) - \right. \\
& \left. \phi_{\alpha,\beta}(\boldsymbol{\xi}_{\alpha',m_y-1/2})\boldsymbol{F}_y(\boldsymbol{W}^L_{\alpha',m_y-1/2}, \boldsymbol{W}^R_{\alpha',m_y-1/2})\right),
\end{aligned} \tag{7.36}$$

where the subscripts $m_\gamma \pm 1/2$ ($\gamma = x, y$) refer to the positions in the cell edges, for example $\boldsymbol{\xi}_{\alpha',m_y \pm 1/2} = (\xi_{\alpha'}, \pm 1/2)$ and $\boldsymbol{W}_{\alpha,m_y \pm 1/2} = \boldsymbol{U}_h(\boldsymbol{\Xi}(\boldsymbol{\xi}_{\alpha',m_y \pm 1/2}))$, etc. The two superscripts L and R are used to denote respectively, the left and right state solution with respect to the corresponding edge. Finally, this system can be written as an ODE system of the unknown vector of degrees of freedom:

$$\frac{d\boldsymbol{W}^m}{dt} = RHS(\boldsymbol{W}^m, \boldsymbol{W}^{m,N}, \boldsymbol{W}^{m,W}, \boldsymbol{W}^{m,S}, \boldsymbol{W}^{m,E}), \tag{7.37}$$

where $\boldsymbol{W}^{m,e}$ is the vector of degrees of freedom in the neighbor cell which shares edge e with cell m. This ODE can be integrated with a time integrator of order $p = k+1$, in order to obtain a global numerical scheme of the same order. In the present work, we use a $k+1$-steps Runge-Kutta method Gottlieb et al.(2001).

7.3.3 Positivity limitation

In this section, we explain how we can preserve the realizability constraint on the moments. For this purpose, we use the same idea as the one proposed by Zhang et al.(2012). This method has already been used for the Euler equation. The method can be also used, in general, to preserve convex constraint of the solution of hyperbolic equations. The moment space is a convex set, which allows to apply this method to the present system of equations.

Since the Lagrangian basis function sum to 1 within one cell, by summing over all the degrees of freedom in equation (7.28), and using the forward Euler discretization in time, we obtain that the scheme followed by the mean value in a cell reads:

$$\overline{\boldsymbol{W}}^{m,n+1} = \overline{\boldsymbol{W}}^{m,n} - \frac{\Delta t}{\Delta x} \sum_{e \in D} \int_{\xi=-1/2}^{1/2} \boldsymbol{F}\left(\boldsymbol{U}_h^L(t, \widetilde{\boldsymbol{x}}_e(\xi)), \boldsymbol{U}_h^R(t, \widetilde{\boldsymbol{x}}_e(\xi); \boldsymbol{n}_{m,e})\right) d\xi, \tag{7.38}$$

where $\overline{\boldsymbol{W}}^{m,n} = \int_{\boldsymbol{x} \in \mathcal{K}_m} \boldsymbol{U}_h(t_n, \boldsymbol{x}) d\boldsymbol{x}$ is the mean solution in cell m at $t = t_n$.

The main idea of Zhang & Shu consists in using a quadrature rule which solves for the mean solution in the cell exactly. Using adequate quadrature points, the authors showed that the updated mean solution (7.38) can then be written as a convex combination of the solution at these quadrature points. Thus, the updated mean solution is realizable. This result is valid when we use a realizability preserving numerical flux as the one proposed in (7.15) and an adequate CFL condition. In 2D, Zhang & Shu used two sets of quadrature points in $[-1/2, 1/2]^2$ given by the Gauss-Legendre-Lobbato (GLL) quadrature abscissas:

$$\begin{aligned} S_x &= \{(\xi_\alpha, \hat{\xi}_r) : 1 \le \alpha \le k+1, 1 \le r \le k+2\} \\ S_y &= \{(\hat{\xi}_s, \xi_\beta) : 1 \le r \le k+2, 1 \le \beta \le k+1\}, \end{aligned} \tag{7.39}$$

where $(\hat{\xi}_r)_{1 \le r \le k+2}$ is the sequence of 1D Gauss-Lobbato abscissas of order $k+2$. The main result of Zhang & Shu work can be summarized in the following proposition:

Proposition 7.3.1 *Let* $\mathcal{G}$ *be a convex set, and we suppose that for all* $\boldsymbol{\xi} \in S_x \cup S_y$ *we have* $\boldsymbol{U}_h(t_n, \Xi_m(\boldsymbol{\xi})) \in \mathcal{G}$ *and the resolution of (7.36) is achieved with a realizability preserving numerical flux, then the mean solution at* $t_{n+1} = t_n + \Delta t$ *belongs to the convex set* $\mathcal{G}$, $\overline{\boldsymbol{W}}^{m,n+1} \in \mathcal{G}$ *under the following CFL condition:*

$$\frac{\Delta t}{\Delta x} = CFL \frac{\hat{w}_1}{\max(|\lambda_l^x|) + \max(|\lambda_l^y|)} \tag{7.40}$$

where $CFL \le 1$, $\hat{w}_1$ *is the first Gauss-Labotto weight of order* $k+2$ *and* λ_l^x *(respectively* λ_l^y*) are the eigenvalues of the hyperbolic system in the direction* x *(respectively* y*). In the present case, system (7.1) is weakly hyperbolic: it has one eigenvalue in each direction* $\lambda_l^x = u$ *and* $\lambda_l^y = v$.

We underline that the above proposition can be generalized to the case of other Strong Stability Preserving (SSP) time integrator Gottlieb et al.(2001).

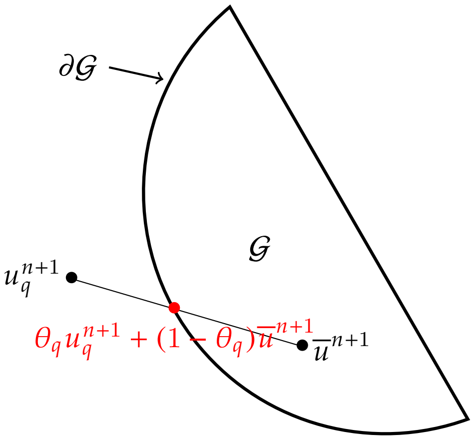


Figure 7.3: *Illustration of the limitation procedure: the solution associated to a GLL quadrature point* $\boldsymbol{U}_q^{n+1} = \boldsymbol{U}_h(t_{n+1}, \boldsymbol{x}_q)$ *is lying outside the space of constraints, the red point shows its projection on the border of this space.*

In the following, we define a convex set G_ϵ parametrized by a small variable ϵ:

$$\mathcal{G}_\epsilon = \{\boldsymbol{U} = (\mathbf{m}^t, \mathrm{m}_1\boldsymbol{u}^t)^t : \mathbf{c}(\mathbf{m}) \in \mathbb{M}_3^{1/2}, \text{ and } \boldsymbol{p}(\mathbf{c}) \in [\epsilon, 1-\epsilon]^3\} \tag{7.41}$$

where $\mathbf{c}(\mathbf{m}) = (\mathrm{m}_{1/2}/\mathrm{m}_{0/2}, \mathrm{m}_{2/2}/\mathrm{m}_{0/2}, \mathrm{m}_{3/2}/\mathrm{m}_{0/2})$ is the normalized fractional moment associated to the moment vector $\mathbf{m}$ and $\boldsymbol{p}(\mathbf{c})$ is the canonical fractional moment defined in 4.4.3. The parameter $\epsilon > 0$ is chosen by the user to be strictly positive to force the moment vector to stay in the interior of the moment space and keep a strictly distance from the moment space border. Indeed, for a moment vector in or very close to the moment space border, the reconstruction of the NDF through entropy maximization encounters some numerical difficulties, see Vié et al.(2013b). In practice, we set $\epsilon = 1.e^{-4}$. For the resolution of the numerical scheme, we use a family of SSP Runge-Kutta method Gottlieb et al.(2001) and the realizability preserving upwind numerical flux, defined in (7.15). So far, the only remaining point is to ensure the realizability of the solution at GLL points:

$$\Big(\forall \boldsymbol{x} \in \Xi_m(S_x \cup S_y)\Big), \; \boldsymbol{U}_h(t, \boldsymbol{x}) \in \mathcal{G}_\epsilon, \tag{7.42}$$

at each time step. Indeed, the initial condition is supposed to be physical and thus realizable everywhere, in particular at the quadrature points. By using a realizability-preserving numerical flux, SSP time integrator and keeping the restricted CFL condition, we obtain realizable mean values in each cell for the first step. But nothing ensures the realizability at the GLL points. Therefore, we use a limitation procedure, after each time step, to bring the solution at GLL points back in the convex space $\mathcal{G}_\epsilon$. Let us suppose that at the time t_n the mean solution is in the space $\mathcal{G}_\epsilon$. We denote by $(\boldsymbol{x}_q)_{1\leq q\leq 2(k+1)(k+2)}$ a sequence of points of $\Xi_m(S_x \cup S_y)$. Since $\mathcal{G}_\epsilon$ is a convex set, there exists a unique $\theta_q \in [0,1]$, so that $\widetilde{\boldsymbol{U}}_h(t_n, \boldsymbol{x}_q) = \theta_q \boldsymbol{U}_h(t_n, \boldsymbol{x}_q) + (1-\theta_q)\overline{\boldsymbol{W}}^{m,n}$ lies on the boundary of G_ϵ, see Figure 7.3. The numerical solution is then corrected as follows:

$$\widetilde{\boldsymbol{W}}_{\alpha,\beta}^{m,n+1} = \theta \boldsymbol{W}_{\alpha,\beta}^{m,n+1} + (1-\theta)\overline{\boldsymbol{W}}^{n+1}, \quad \theta = \min_{1\leq q\leq 2(k+1)(k+2)}(\theta_q) \tag{7.43}$$

This limitation procedure preserves the mean cell value, therefore the numerical method remains conservative and it is shown in Zhang and Shu(2011) that the accuracy of the scheme is not impacted in the case of regular solutions. The solution $\widetilde{\boldsymbol{U}}_h(t, \boldsymbol{x})$ is then $(k+1)^{th}$ order accurate and realizable at all GLL quadrature nodes.

7.3.4 Maximum principle on the velocity

PGD systems, like system (7.1), can lead to δ-shocks. High order numerical methods often fail in such critical situations, where the numerical resolution leads to high oscillations and violate the maximum principle on the transported variables and particularly on the velocity. In order to cope with this issue, we propose to use a limitation method on the velocity by using a procedure similar to the one used to ensure the realizability of the moments. The objective here is to satisfy the maximum principle on the velocity for the mean solution $\overline{\boldsymbol{W}}^{m,n}$. Numerically, this can be written as follows:

$$\begin{aligned}\min_{m'\in O(m)}\left(\overline{u}^{m',n}\right) \leq \quad \overline{u}^{m,n+1} \quad &\leq \max_{m'\in O(m)}\left(\overline{u}^{m',n}\right),\\ \min_{m'\in O(m)}\left(\overline{v}^{m',n}\right) \leq \quad \overline{v}^{m,n+1} \quad &\leq \max_{m'\in O(m)}\left(\overline{v}^{m',n}\right),\end{aligned} \tag{7.44}$$

where $O(m)$ is the set of neighbor cells of m, including cell m itself.

In the following, we denote by $\mathcal{M}^{m,n}$ to the set of all possible solutions $(\overline{u}^{m,n+1}, \overline{u}^{m,n+1})$ satisfying the conditions in (7.44) for a given time step t_n and cell m. This set is convex, we can then apply the procedure of Zhang & Shu in the same way as presented in the last section. We summarize the limitation procedure to preserve the realizability of moments and the maximum principle on the velocity as follows:

- We first compute the solution at $t = t_{n+1}$ at the GLL points:

$$\boldsymbol{U}_h(t_n, \boldsymbol{x}_q) = \sum_{1\leq\alpha,\beta\leq k+1} \boldsymbol{W}_{\alpha,\beta}\phi_{\alpha,\beta}(\Xi_m^{-1}(\boldsymbol{x}_q)). \tag{7.45}$$

- We determine the maximum $\theta \in [0, 1]$ such that for all $1 \leq q \leq 2(k+1)(k+2)$, we have:

$$\theta \boldsymbol{W}(t_{n+1}, \boldsymbol{x}_q) + (1-\theta)\overline{\boldsymbol{W}}^{n+1} \in \mathcal{G}_\epsilon. \tag{7.46}$$

- We determine the maximum $0 \leq \vartheta \leq \theta$ such that for all $1 \leq q \leq 2(k+1)(k+2)$, we have:

$$\vartheta \boldsymbol{W}(t_{n+1}, \boldsymbol{x}_q) + (1-\vartheta)\overline{\boldsymbol{W}}^{n+1} \in \mathcal{M}^{m,n}. \tag{7.47}$$

- Finally, we limit the solution as follows:

$$\widetilde{\boldsymbol{W}}^{m,n+1}_{\alpha,\beta} = \vartheta \boldsymbol{W}^{m,n+1}_{\alpha,\beta} + (1-\vartheta)\overline{\boldsymbol{W}}^{n+1}. \tag{7.48}$$

7.4 Numerical tests and validation

This section is dedicated to some representative 1D test cases and analysis of numerical results, to verify the robustness and the accuracy of the two transport schemes (KFV and RKDG). First, we investigate the accuracy of the two schemes using different order of accuracy on a simple regular advection case. Then, we continue the accuracy study in a more complex case, which presents high gradients of the solution. In this test case, we compare and analyze the two scheme classes. Finally, a critical δ-shock case is performed to evaluate the robustness of the two schemes. For the three test cases, we consider a periodic domain $[0, 1]$ and we use a $CFL = 0.75$, where the CFL condition for KFV is defined in (7.16)-(7.25), and for RKDG is defined in (7.40). Multi-dimensional simulations are not presented in this chapter. The reader can find 2D and 3D simulations in chapters 8, 9 and 10.

7.4.1 Advection test case

In this section, we consider a simple advection with a constant velocity $u = -1$. The initial moments are presented in Figure 7.4 and are expressed as follows:

$$m_{k/2}(t=0,x) = \frac{2}{k+2}(S_{max}^{(k+2)/2} - S_{min}^{(k+2)/2})\exp(-(x-x_c)^2/\sigma_x^2) \tag{7.49}$$

where $(S_{min}, S_{max}) = (0.3, 0.7)$, $x_c = 0.5$ and $\sigma_x = 0.1$. The initial moments correspond to the moments of a rectangular spray size distribution $n(t,S) = \mathbb{1}_{[S_{min},S_{max}]}(S)$ multiplied by a Gaussian spatial distribution.

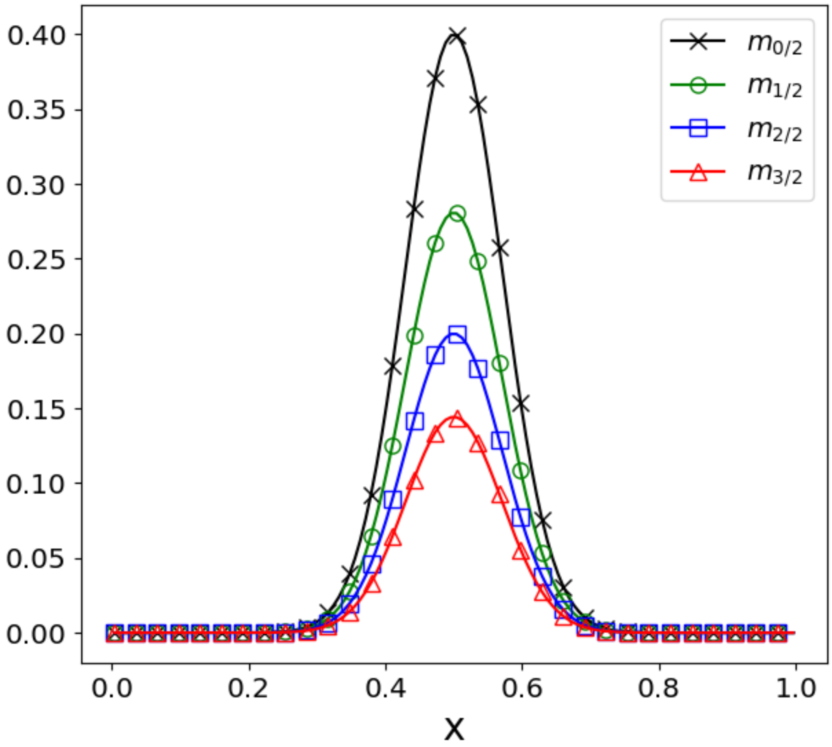


Figure 7.4: *Initial moments for the advection case: $m_{0/2}$ (cross), $m_{1/2}$ (circle), $m_{2/2}$ (square) and $m_{3/2}$ (triangle).*

In Figure 7.5, we compare the numerical solutions of KFV (order 1 and 2) with RKDG (order 1, 2 and 3). The first order solution of both schemes leads to a high numerical diffusion, while the second order is accurate enough to preserve the spatial profile after crossing twice the domain at $t = 2$. However, the second order RKDG scheme is more accurate, such that we can not identify the difference with the exact solution. Figure 7.6 gives more details of the convergence order of the different schemes. In the logarithmic scale, the L^1-errors of the solution obtained with the second order RKDG decreases asymptotically with a slope of -2.5 and the third order RKDG with a slope -3.3, while for the second order KFV scheme the errors decreases only with a slope of -1.55 as we increase $-\ln(\Delta x)$.

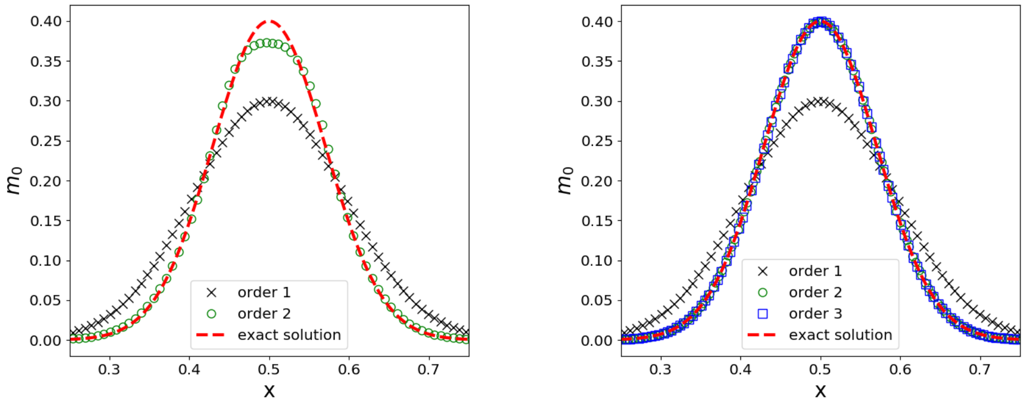


Figure 7.5: *Number density for the advection case at* $t = 2$*. Left: using KFV scheme of order* 1 *(cross) and order* 2 *(circle). Right: using RKDG scheme order* 1 *(cross), order* 2 *(circle) and order* 3 *(square).*

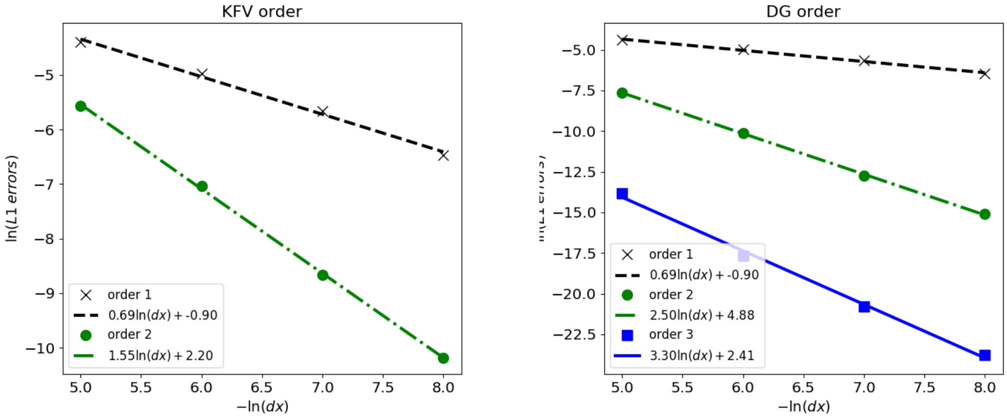


Figure 7.6: *Error curves of* m_0 *with respect to grid refinement in logarithm scale. Left: KFV schemes. Right: RKDG schemes. First order (cross) and second order (circle) for the two scheme types and third order (square) only for RKDG scheme.*

7.4.2 Compression test case

We consider an initial size distribution with a form depending on the coordinate space x. In fact, we need a non trivial initial size-space distribution, such that the slopes used in the second order scheme of the FKV method do not vanish all the time. The chosen initial size distribution has the following profile:

$$n(t=0,S,x) = 10\exp\left(-\frac{(x-x_c)^2}{\sigma_x^2}\right)\exp\left(-\frac{(\sqrt{S}-(1-x)/2)^2}{\sigma_R^2}\right), \tag{7.50}$$

where $x_c = 0.25$, $\sigma_x = 0.1$ and $\sigma_R = 0.3$. The initial velocity field is initiated as follows:

$$\left\{\begin{array}{lcll} u(t=0,x) & = & 0.5-x & x<0.5, \\ u(t=0,x) & = & 0. & x \geq 0.5. \end{array}\right. \tag{7.51}$$

Figure 7.7 shows the initial conditions for the fractional moments and the velocity. We can obtain an analytic solution of this test case using the characteristic curves method. For $t < 1.0$, the characteristic curves do not cross as shown in Figure 7.8, and we can show that the NDF solution of the transport equation (7.10), can be written as follows:

$$\begin{array}{rcl} n(t,S,x) & = & \dfrac{1}{1.-t}\,n(0,S,x_o(t;x)) \\ x_o(t;x) & = & \dfrac{x-0.5t}{1.-t} \end{array} \tag{7.52}$$

At $t = 1.0$ (see Figure 7.8), all the characteristic curves cross at point $(t = 1.0, x = 0.5)$, what leads to a δ-Dirac distribution in space. In the following, we study the numerical accuracy of the two methods KFV and RKDG. We investigate the numerical solution for $t \leq 0.8$, before the formation of the shock. In Figures 7.9-7.10, we display the solution at two different times ($t = 0.4$ and $t = 0.8$) and we use the two method using different orders. The first order version of the two schemes smears out the solution, in the same way, due to a high numerical diffusion. Furthermore, it is simple to show that the first order KFV scheme is equivalent to the RKDG one. Nonetheless, the solutions obtained with the two methods higher order resolution are different. We can first remark the gain of accuracy brought by the second order RKDG and KFV in comparison to the first order scheme. But the numerical solution obtained with the RKDG second order scheme is more accurate than the one obtained with KFV second order scheme. To go further in this analysis, we compute the L^1−error at $t = 0.8$ for each numerical schemes and for different orders depending on the grid size. In Figure 7.11, we display the convergence curves, highlighting the difference of precision between the two methods (KFV and RKDG). As it can be seen from these curves, the order accuracy of the KFV is much lower than the theoretical order. Indeed, the order of accuracy of the first order is around 0.4 and the second order is only 1.3. The low resolution of KFV is due to the numerical diffusion, which is present naturally for the first order, while for the second order, it is introduced by the slope limitation procedure. Indeed, this is a difficult test case, where the solution starts to have high gradient variations as we can see in Figure 7.9. The slope limitation procedure ensures the realizability of the moments and the maximum principle on the velocity and canonical moments. But has a cost in terms of accuracy. However, RKDG second and third order seem not to be so affected by the Zhang & Shu limitation procedure. For the second order RKDG, the logarithmic of $L^1 - error$ of the solution decreases with a slope of 2.02 and the third order RKDG with a slope of 2.64 depending on $-\ln(dx)$.

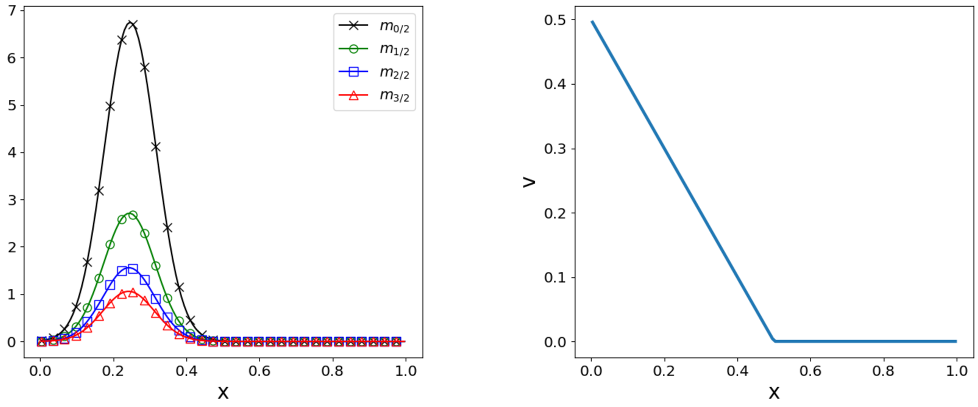


Figure 7.7: *Initial condition for the compression test case. Left: Initial moment fields, the curves represent the moment with decreasing order in terms of value. Right: initial velocity*

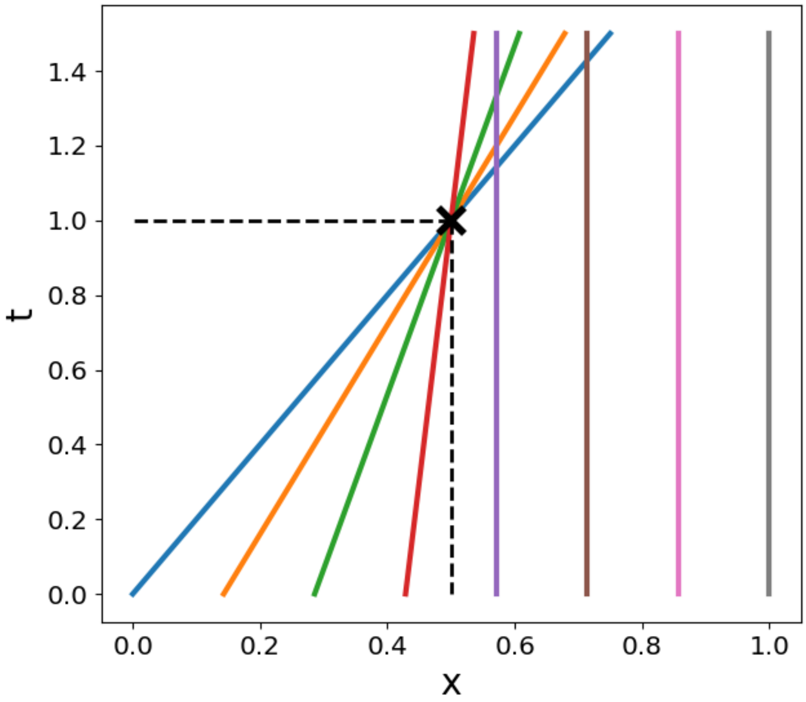


Figure 7.8: *Illustration of the characteristic curves of the solution corresponding to the second test case. Remind that the equation on the velocity is a Burger's equation.*

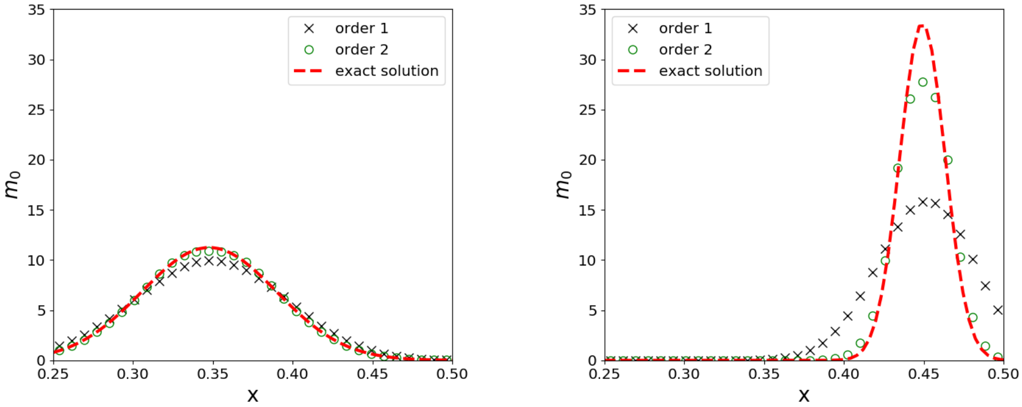


Figure 7.9: *Number density for the compression test case at* $t = 0.4$ *(left) and* $t = 0.8$ *(right). Solution obtained with the KFV scheme of order* 1 *(cross) and order* 2 *(circle).*

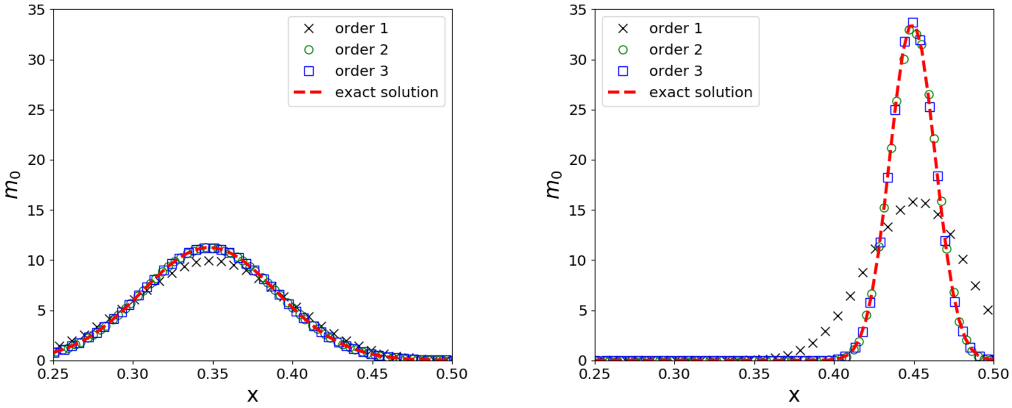


Figure 7.10: *Number density for the compression test case at* $t = 0.4$ *(left) and* $t = 0.8$ *(right). Solution obtained with the RKDG scheme of order* 1 *(cross), order* 2 *(circle) and order* 3 *(square).*

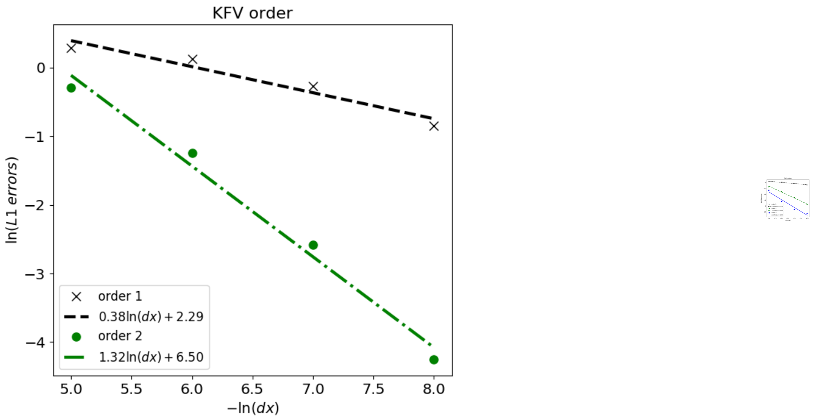


Figure 7.11: *Error curves of* m_0 *with respect to grid refinement in logarithm scale. Left: KFV schemes and on the right: RKDG schemes. First order (cross) and second order (circle) for the two scheme types and third order (square) only for RKDG scheme.*

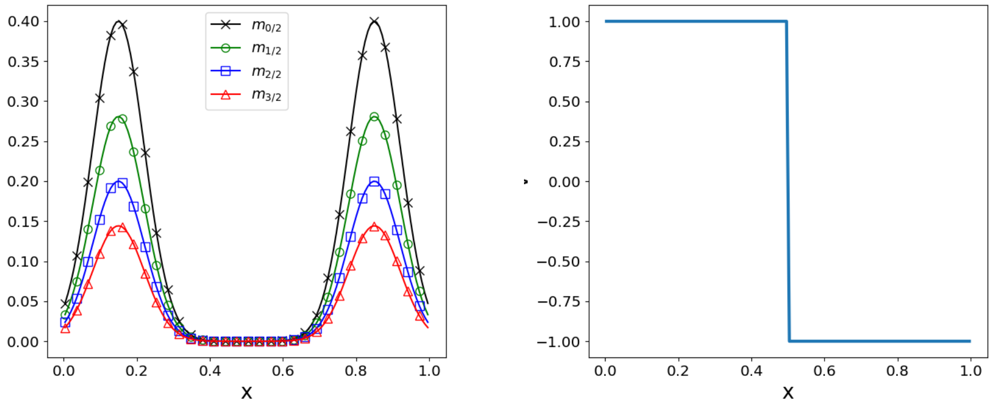


Figure 7.12: *Initial condition for the crossing case. Left: Initial moment fields, the curves represent the moment with decreasing order in terms of value. Right: initial velocity.*

7.4.3 Robustness and capacity of capturing δ-shocks

The monokinetic-assumption is not more valid hypothesis when droplets cross. In case of such an event, the monokinetic-model generates a δ-shock. Despite the non-physical solution, the two schemes should be able to run in all critical situations and even resolve singularities. In this section, we test the robustness of the numerical schemes in the case of a strong crossing. Initially, we consider two spatial Gaussian distributions centered at two different symmetric positions and traveling at opposite velocity, as shown in Figure 7.12.

Figures 7.13-7.14 show the spatial profile of the moment m_0 at two different times ($t = 0.25$ and $t = 0.5$). We can see the shock generation when the two packets cross. At the end, the droplets accumulate at the center and the corresponding spacial distribution has the form of a delta distribution. We compare the solutions with an exact solution, which is represented in Figure 7.13-7.14, in Finite Volume cells[2]. It is not obvious to compare the accuracy of the two schemes for different orders due to the presence of a Dirac-Delta function. However, the main objective of this test case is to evaluate the robustness of high order schemes (second order for KFV and second and third orders for RKDG) in such critical situations. Second order KFV is more robust compared to high order RKDG schemes. Indeed, the scheme is designed in such way to respect the maximum principle of both the velocity and the canonical moments. For the second and the third order RKDG schemes, we use the limitation procedure described in the section 7.3.4, in order to satisfy the maximum principle on the velocity and thus to obtain a stable solution. The method works well for the second order, however for the third order we have needed to modify the parameter ϑ given in (7.47) to end up with a stable solution. In this simulation, we multiply ϑ by 0.9 to obtain the solution in Figure 7.14. In future work, we will need to investigate further the limitation procedure used to respect the maximum principle on the velocity and eventually to use a third limitation to ensure the maximum principle on the canonical moments.

---

[2] Dirac-function is represented by the function $1./dx\mathbb{1}_{[0.5-dx/2.,0.5+dx/2.]}(x)$.

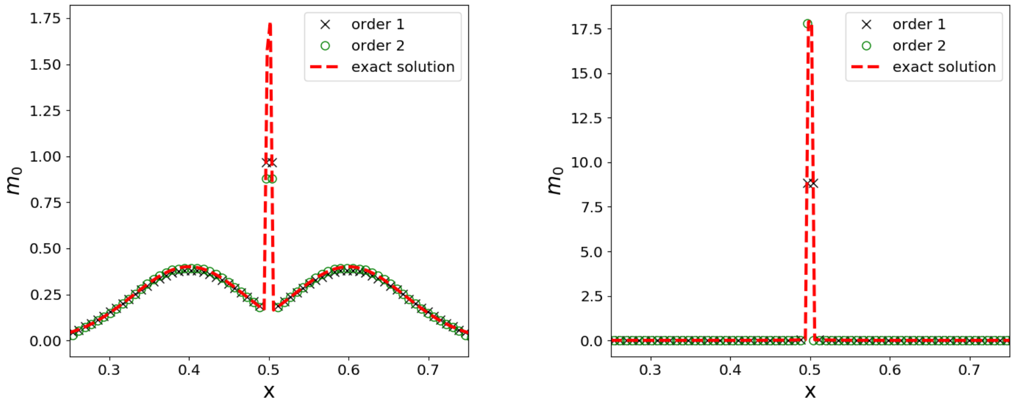


Figure 7.13: *Number density for the crossing test case at* $t = 0.25$ *(left) and* $t = 0.5$ *(right). Solution obtained with KFV scheme of order* 1 *(cross) and order* 2 *(circle).*

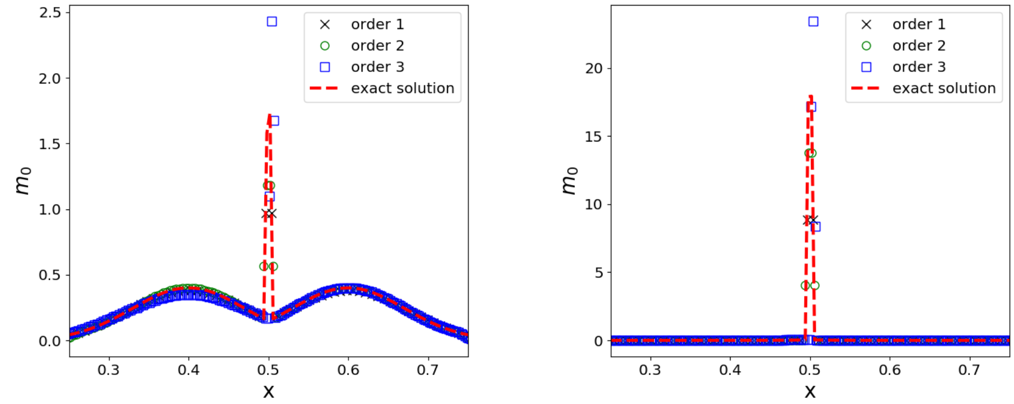


Figure 7.14: *Number density for the crossing test case at* $t = 0.4$ *(left) and* $t = 0.8$ *(right). Solution obtained with KFV scheme of order* 1 *(cross) and order* 2 *(circle).*

7.5 Conclusion

This chapter is dedicated to high order and realizable numerical schemes to solve the transport part of the fractional moment model presented in chapter 4. Two types of schemes are presented during this chapter. The first one is based on a finite volume discretization. The numerical flux is obtained by using the equivalence between the fractional moments PGD-like system and kinetic equation for a monokinetic distribution in the velocity. We show how we can obtain first and second order schemes, while ensuring the realizability of the moments and also satisfying the maximum principle on the velocity and on the canonical moments. We refer to this type of scheme by Kinetic Finite Volume (KFV) schemes. The second type is the Runge Kutta Galerkin Discontinuous (RKDG) scheme. The method can be extended to an arbitrary order of accuracy. The realizability of the moments is ensured by using the limitation procedure proposed the first time by Zhang & Shu. We have also used this limitation technique to ensure the maximum principle on the velocity. In fact, the maximum principle on the velocity allows to obtain a stable solution, especially in the critical case of δ-shocks formation. However, we still have some open issues to ensure the robustness of high order RKDG schemes in such cases. Indeed, we show in the last test case that we need to tune the limitation coefficient to obtain a stable solution for the third order RKDG scheme. On the other side, KFV shows a robust behavior in this critical situation. Finally, we show for other test cases that we obtain accurate solutions using high order RKDG schemes compared to KFV schemes. Furthermore, the accuracy orders obtained with the RKDG are very close to the theoretical ones. While for KFV schemes, we find that the second order improve the accuracy compared to the first order, but the accuracy order is lower than the theoretical order because of the slope limitation.

Chapter 8

A new algorithm for the time integration of the source terms in the fractional moments model

8.1 Introduction

This chapter presents a numerical procedure for the resolution of the source terms of the fractional moments model derived in chapter 4. Recall that the numerical resolution of the transport part has been treated in chapter 7. The resolution of the source terms turns to a time integration of an ODE system. However, classical ODE solvers do not ensure the realizability of the moments. In the present work, we propose a new realizable algorithm to solve the evolution of the fractional moments due to evaporation. The resolution of the evaporation is first done by evaluating the droplets disappearance flux. The reconstruction through Maximum Entropy formalism is used here to evaluate this flux. Then, we compute the internal size evolution by using a specific quadrature, which involves negative order moments and requires an original strategy compared to the integer moment problem Massot et al.(2010). Indeed, we show that the evaporation algorithm proposed for the EMSM model is not accurate to evaluate correctly the evaporation in the case of the fractional moments model. The proposed strategy is then assessed by a careful investigation of the numerical errors, as well as a detailed comparison with analytic solutions in 0D test cases. Finally, we compare the EMSM and the fractional moments model on 2D acedemic configuration. In this last simulation, we conduct a complete resolution of the system of equations (4.24) including the transport and the evaporation and the drag force source terms.

8.2 Resolution of the evaporation

Let us consider a pure evaporation, without transport nor drag. The kinetic equation in this case reads:

$$\partial_t n - \partial_S(Kn) = 0. \tag{8.1}$$

The equations corresponding to the fractional moments model with a continuous reconstruction of the NDF through Entropy maximization reads:

$$\begin{cases} d_t \mathrm{m}_{0/2} &= -\mathrm{K}n^{ME}|_{S=0}, \\ d_t \mathrm{m}_{1/2} &= -\dfrac{\mathrm{K}}{2}\mathrm{m}_{-1/2}, \\ d_t \mathrm{m}_{2/2} &= -\mathrm{K}\mathrm{m}_{0/2}, \\ d_t \mathrm{m}_{3/2} &= -\dfrac{3\mathrm{K}}{2}\mathrm{m}_{1/2}, \end{cases} \tag{8.2}$$

where the instantaneous disappearance flux for zero size, $\mathrm{K}n^{ME}|_{S=0}(\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, \mathrm{m}_{2/2}, \mathrm{m}_{3/2})$ is obtained through an Entropy Maximization detailed in section 4.4.5. The RHS of this ODE system corresponds to the evaporation source term of the system of equations (4.24).

Solving this system using classical integrator such as Euler or Runge-Kutta methods does not ensure the realizability of the moments Massot et al.(2010). This property is essential for robustness and accuracy to reconstruct a positive NDF. For this reason, we design a new algorithm to solve the evaporation based on the kinetic evolution of the NDF, given by equation (8.1). First, we present a simple method to compute directly the updated moments from the exact kinetic solution of the NDF. This method is not well suitable for real spray applications with complex evaporation laws, because it requires intense arithmetic computations. However, we use this method as a reference solution. The evaporation algorithm used for the EMSM model can be used for general evaporation law with less arithmetic operations. We will show in a second part that this algorithm is inaccurate when evaluating the evaporation in the case of the fractional moments model. Finally, we highlight the main problem of this algorithm and we propose a new strategy to cope with this difficulty.

8.2.1 Exact kinetic solution through the method of characteristics

The exact solution of the NDF evolution, in the case of d^2 evaporation law, can be obtained easily by solving analytically the kinetic equation (8.1):

$$n(t, S) = n(0, S + \mathrm{K}t). \tag{8.3}$$

For more general evaporation law, when the evaporation rate $\mathrm{R}_S(S)$ is a smooth function of the size, the kinetic equation can be solved by using the method of characteristics. Indeed, by multiplying the kinetic equation by $\mathrm{R}_S(S)$, we obtain the following equation in the function $\Gamma(t, S) = \mathrm{R}_S(S)n(t, S)$:

$$\partial_t \Gamma(t, S) + \mathrm{R}_S(S)\partial_S(\Gamma(t, S)) = 0. \tag{8.4}$$

Then, for a given initial time t_0 and size S_o, we define the one variable function $g(t) = \Gamma(t, \widetilde{\phi}(t; t_0, S_o))$, such that $\widetilde{\phi}(t; t_0, S_o)$ is the characteristic curve verifying:

$$\begin{cases} \dfrac{d\widetilde{\phi}(t; t_0, S_o)}{dt} &= \mathrm{R}_S(\widetilde{\phi}(t; t_0, S_o)), \\ \widetilde{\phi}(t_0; t_0, S_o) &= S_o. \end{cases} \tag{8.5}$$

It is a simple fact that the derivative of $g(t)$ vanishes, and thus, we obtain the following expression:

$$\Gamma(t, \widetilde{\phi}(t; t_0, S_o)) = \Gamma(t_0, S_o). \tag{8.6}$$

Finally, we obtain the exact solution of the size distribution as follows:

$$n(t,S) = \frac{\mathrm{R}_S(\widetilde{\phi}(t_0;t,S))}{\mathrm{R}_S(S)} n(t_0, \widetilde{\phi}(t_0;t,S)). \tag{8.7}$$

8.2.2 Fully kinetic scheme

At each time step t_n, the reconstructed NDF $n^{ME}(t = t_n, S)$ is determined using the ME algorithm. The exact kinetic solution of the NDF can then be expressed analytically as a function of this initial NDF thanks to (8.3), for a d^2 evaporation law. Finally, the updated moments are computed as follows:

$$\begin{aligned}
\mathrm{m}_{k/2}(t_n + \Delta t) &= \int_0^1 S^{k/2} n(t_n + \Delta t, S) dS \\
&= \int_0^1 S^{k/2} n(t_n, S + \mathrm{K}\Delta t) dS \\
&= \int_0^{1-\mathrm{K}\Delta t} S^{k/2} n(t_n, S + \mathrm{K}\Delta t) dS \\
&= \int_{\mathrm{K}\Delta t}^1 (S - K\Delta t)^{k/2} n^{ME}(t_n, S) dS.
\end{aligned} \tag{8.8}$$

In these equalities, we consider $n(t, S \geq 1) = 0$, since the dimensionless maximum size is $S_{max} = 1$.

The integral calculation can be achieved by using a Gauss-Legendre quadrature. The method is simple and accurate. However, its extension to complex evaporation laws, where the evaporation rate depends on several parameters, could involve heavy calculations. To clarify this point, we consider a smooth evaporation rate $\mathrm{R}_S(S)$, function of the droplet size. Using the solution (8.7), we can that the updated moments can be expressed as follows:

$$\mathrm{m}_{k/2}(t_{n+1}) = \int_{\widetilde{\phi}(t_n;t_{n+1},0)}^1 \widetilde{\phi}(t_{n+1};t_n,s)^{k/2} n(t_n,s) ds. \tag{8.9}$$

To compute accurately this integral, we use 24 Gauss quadrature points. However, we also need to compute the size $\widetilde{\phi}(t_{n+1}, t_n, S_j)$, for all abscissas S_j of the 24-quadrature points and $\widetilde{\phi}(t_n; t_{n+1}, 0)$, by solving the ODE system (8.5). For this reason, we consider this direct method in solving the moments will be very costly in terms of CPU time.

8.2.3 Inefficiency of the original EMSM algorithm for evaporation

Massot et al Massot et al.(2010) proposed a realizable algorithm to solve correctly the evaporation moment system in the case of integer moments. The idea of this algorithm is also based on the known solution of the kinetic equation (8.1). In order to design a generalized algorithm for a d^2 evaporation law or for more complex laws, while keeping low computational cost, the calculation of the moments is done in three steps instead of computing directly the integrals (8.13) or (8.9) using a large number of Gauss-Legendre quadratures. This algorithm reads in the case of fractional moments as follows:

- From the moment vector $\mathbf{m}(t_n) = (\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, \mathrm{m}_{2/2}, \mathrm{m}_{3/2})^t$, we reconstruct the NDF by using the ME algorithm. Then, we compute the disappearance flux of the droplets which will be totally evaporated at the next step:

$$\mathbf{\Phi}_{-}(t_n) = \int_0^{\mathrm{K}\Delta t} n_{ME}(t_n, s) \begin{pmatrix} 1 \\ s^{1/2} \\ s \\ s^{3/2} \end{pmatrix} ds. \tag{8.10}$$

- Using the Product-Difference (PD) algorithm Gordon(1968), we compute the abscissas $S_j \geq \mathrm{K}\Delta t$ and the weights $w_j \geq 0$ of the lower principal representation of the moments[1] $\mathbf{m}^{[\mathrm{K}\Delta t,1]}(t_n) = \mathbf{m}(t_n) - \mathbf{\Phi}_{-}(t_n)$. This corresponds to solving the system:

$$\mathrm{m}_{k/2}^{[\mathrm{K}\Delta t,1]}(t_n) = \sum_{j=1}^{2} w_j S_j^{k/2}, \quad k \in \{0, 1, 2, 3\}, \tag{8.11}$$

  where $\mathbf{m}^{[\mathrm{K}\Delta t,1]}(t_n)$ is the moment vector on $[\mathrm{K}\Delta t, 1]$.

- Finally, we calculate the moments at the next step:

$$\mathrm{m}_{k/2}(t_n + \Delta t) = \sum_{j=1}^{2} w_j (S_j - \mathrm{K}\Delta t)^{k/2}. \tag{8.12}$$

However, this algorithm is not adapted for the fractional moments and can not predict the correct kinetic evolution of the NDF. We illustrate this problem in Figure 8.1. Two solutions of the NDF are constructed through ME algorithm from two solutions of the moments: the first set of moments is computed using the above algorithm and the second one by using the fully kinetic scheme. These two solutions are compared to the exact kinetic solution as proposed in equation (8.3). The NDF obtained with the fully kinetic scheme fits accurately the exact solution. However, the one obtained using the above algorithm shows the inefficiency of this method to predict the right kinetic evolution of the NDF from the fractional moments. In the next section, we give a mathematical explanation of this unexpected failure of the method described above, then we propose a new original solution to adapt the algorithm.

8.2.4 Adapted evaporation scheme for fractional moments

In order to understand the inefficiency of the previous algorithm proposed in section 8.2.3, we propose to consider again a constant evaporation rate $\mathrm{R}_S(S) = -\mathrm{K}$. The moment solution expressed directly from the exact kinetic solution reads:

$$\begin{aligned} \mathrm{m}_{k/2}(t+\Delta t) &= \int_0^1 S^{k/2} n(t+\Delta t, S) \mathrm{d}S = \int_0^1 S^{k/2} n(t, S+\mathrm{K}\Delta t) \mathrm{d}S \\ &= \int_{\mathrm{K}\Delta t}^{1+\mathrm{K}\Delta t} (S-\mathrm{K}\Delta t)^{k/2} n(t,S) \mathrm{d}S = \int_{\mathrm{K}\Delta t}^{1} (S-K\Delta t)^{k/2} n(t,S) \mathrm{d}S. \end{aligned} \tag{8.13}$$

In the previous equality, we consider that $n(t, S) = 0$ for $S \geq 1$. In the following, we show that the updated moments can be written as a function of an infinite set of moments on the support $[\mathrm{K}\Delta t, 1]$.

Lemma 8.2.1 *For all positive integer k and for all $x \in [-1, 1]$ the function $f : x \to (1-x)^{k/2}$ admits a power series which converges normally to the function f:*

$$(1-x)^{k/2} = \sum_{n=0}^{+\infty} a_n^{k/2} x^n, \tag{8.14}$$

---

[1] We can also use hybrid approach with multi-size sections, in this case we take into account the evaporation fluxes coming from the right section: $\mathbf{m}(t_n) - \mathbf{\Phi}_{-}^{(i)}(t_n) + \mathbf{\Phi}_{-}^{(i+1)}(t_n)$, where i indexes the size-section.

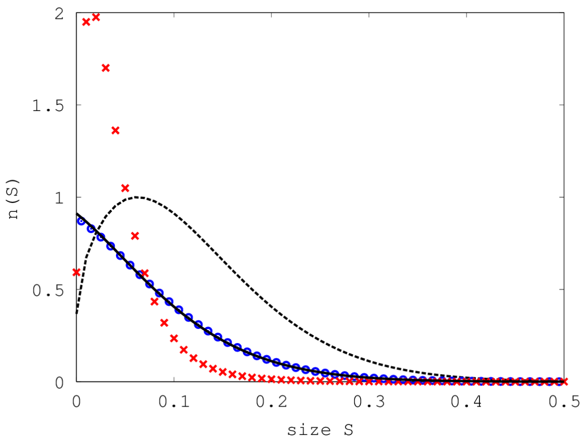


Figure 8.1: *Initial size distribution (dashed line) and the reconstructed size distribution at* $t = 0.2$*: kinetic solution (solid line), EMSM (cross), fully kinetic scheme (circle).*

Proof: The even integer case $k = 2m$ is trivial with $a_n^{k/2} = (-1)^n \binom{m}{n} \mathbb{1}_{[0,m]}(n)$. Let us consider the case where $k = 2m + 1$ is an odd number. The function f admits a power series of the form (8.14). Moreover, its coefficients can be written: $a_0^{k/2} = 1$ and

$$a_n^{k/2} = (-1)^n \frac{(m + 1/2)(m - 1/2)\dots(m + 1/2 - n + 1)}{n!}, \quad n > 0. \tag{8.15}$$

By using the Stirling's approximation, for $n \geq m + 3/2$, we can show the following equivalence relation when n tends to infinity.

$$|a_n^{k/2}| = \left|(-1)^{m+1} \frac{(2(n - m - 1))!(2m + 1)!}{(n - m - 1)!m!2^{2n-1}n!}\right| \sim \frac{(2m + 1)!}{4^m \sqrt{\pi} m!} \frac{1}{n^{3/2+m}}. \tag{8.16}$$

Therefore the series $\sum |a_n^{k/2}|$ is convergent for any integer $k \geq 0$, thus concluding the proof. □

We deduce that $\sum\limits_{n=0}^{\infty} a_n^{k/2} (\mathrm{K}\Delta t)^n s^{k/2-n}$ converges normally to $(s - \mathrm{K}\Delta t)^{k/2}$ for $s \geq \mathrm{K}\Delta t$. Thus, we can invert the sum and the integral in the moment expression:

$$\begin{aligned} \mathrm{m}_{k/2}(t + \Delta t) &= \int_{\mathrm{K}\Delta t}^{1} (s - \mathrm{K}\Delta t)^{k/2} n(t, s) \mathrm{d}s, \\ &= \sum_{n=0}^{\infty} a_n^{k/2} (\mathrm{K}\Delta t)^n \int_{\mathrm{K}\Delta t}^{1} s^{k/2-n} n(t, s) \mathrm{d}s, \\ &= \sum_{n=0}^{\infty} a_n^{k/2} (\mathrm{K}\Delta t)^n \mathrm{m}_{k/2-n}^{[\mathrm{K}\Delta t, 1]}(t). \end{aligned} \tag{8.17}$$

where $\mathrm{m}_{l/2}^{[\mathrm{K}\Delta t,1]} = \int_{\mathrm{K}\Delta t}^{1} s^{k/2-n} n(t, s) \mathrm{d}s$ is a fractional moment of support $[\mathrm{K}\Delta t, 1]$.

Equation (8.17) shows that the fractional moments at $t + \Delta t$ depend on the four fractional moments ($m_l^{[\mathrm{K}\Delta t,1]}$ where $l = 0/2, 1/2 \dots 3/2$) and on an infinite set of the moments of support $[\mathrm{K}\Delta t, 1]$ ($m_l^{[\mathrm{K}\Delta t,1]}$ where $l = -1/2, -2/2, \dots - \infty$). In the case of the EMSM model, where only integer moments are used, the same expansion of the exact kinetic solution of the integer moments involves only the moments of the support $[\mathrm{K}\Delta t, 1]$ and of order $l \in \{0, 1, 2, 3\}$. For this reason, the evolution of the integer moments can be evaluated exactly by translating the abscissas (8.12) of the quadrature (8.11). The same strategy for fractional moments leads to the divergence of the method observed previously. Since we aim at describing the evolution of the moments through the evolution of a minimal number of quadrature nodes, we propose a new quadrature, suitable for fractional moments, such that the updated moments in (8.12) can approximate accurately the exact kinetic evolution (8.17).

8.2.5 A specific quadrature for negative moment order

Our objective is to find an adequate quadrature with the lowest possible quadrature number n_q, such that the following approximation:

$$\mathrm{m}_{k/2}(t+\Delta t) \approx \sum_{j=1}^{n_q} w_j (S_j - K\Delta t)^{k/2}, \tag{8.18}$$

is accurate. More precisely, we would like to find a quadrature such that the difference, for $k \in \{0,1,2,3\}$

$$\begin{aligned} \epsilon_{k/2}(\Delta t) &= \mathrm{m}_{k/2}(t+\Delta t) - \sum_{j=1}^{n_q} w_j (S_j - \mathrm{K}\Delta t)^{k/2} \\ &= \sum_{n=0}^{\infty} a_n^{k/2} (\mathrm{K}\Delta t)^n \left(m_{k/2-n}^{[\mathrm{K}\Delta t,1]}(t) - \sum_{j=1}^{n_q} w_j S_j^{k/2-n} \right), \end{aligned} \tag{8.19}$$

is at least $o(\Delta t)$ to ensure the convergence of the numerical scheme. It is a difficult task to prove such a result, but we provide here a first result is this direction. Our strategy consists in using a quadrature, which cancels a finite set of the first terms in the sum (8.19), by enforcing $E_{p/2} = \mathrm{m}_{p/2}^{[\mathrm{K}\Delta t,1]}(t) - \sum_{j=1}^{n_q} w_j S_j^{p/2}$ to be zero for $p \in \{-2{n_q}^-, \ldots, 2, 3\}$. Thus $2n_q$ fractional moments on $[\mathrm{K}\Delta t, 1]$ are considered, with $n_q = {n_q}^- + 2$, from order $-{n_q}^-$ to $3/2$. As soon as ${n_q}^- \geq 0$, this is sufficient to have $\epsilon_{0/2} = 0$ and $\epsilon_{2/2} = 0$. The following Lemma shows the existence and uniqueness of such quadrature and gives the corresponding value of the error terms $\epsilon_{k/2}(\Delta t)$.

Lemma 8.2.2 *Let* $\left(\mathrm{m}_{p/2}^{[\mathrm{K}\Delta t,1]}\right)_{p\in\{-2{n_q}^-,\ldots,3\}}$ *be a set of fractional moments on* $[\mathrm{K}\Delta t, 1]$ *in the interior of the moment space and* ${n_q}^-$ *a non negative integer. There exists a unique set of abscissas* $(S_j)_{j\in\{1,\ldots,n_q\}}$ *and weights* $(w_j)_{j\in\{1,\ldots,n_q\}}$, *with* $n_q = {n_q}^- + 2$ *such that*

$$\forall k \in \{-2{n_q}^-, \ldots, 3\} \qquad \mathrm{m}_{k/2}^{[\mathrm{K}\Delta t]} = \sum_{j=1}^{n_q} w_j S_j^{k/2}. \tag{8.20}$$

Moreover one have $w_j = w'_j r_j^{2{n_q}^-}$ *and* $S_j = r_j^2$, *where* r_j *and* w'_j *are the abscissas and weights of the Gauss quadrature corresponding to the moments* $m_k = \mathrm{m}_{k/2+{n_q}^-}^{[\mathrm{K}\Delta t,1]}$ *for* $k \in \{0, \ldots, 2n_q - 1\}$. *Using this quadrature, the error terms defined by* (8.19) *can be written:*

$$\begin{aligned} \epsilon_{0/2}(\Delta t) &= 0, \quad \epsilon_{1/2}(\Delta t) = \left(\sum_{n=n_q-1}^{+\infty} a_n^{1/2} \bar{E}_{(2n-1)/2} \right) (\mathrm{K}\Delta t)^{1/2}, \\ \epsilon_{2/2}(\Delta t) &= 0, \quad \epsilon_{3/2}(\Delta t) = \left(\sum_{n=n_q}^{+\infty} a_n^{3/2} \bar{E}_{(2n-3)/2} \right) (\mathrm{K}\Delta t)^{3/2}, \end{aligned} \tag{8.21}$$

where $\bar{E}_{p/2} = (\mathrm{K}\Delta t)^{p/2} \left(\mathrm{m}_{-p/2}^{[\mathrm{K}\Delta t,1]}(t) - \sum_{j=1}^{n_q} w_j S_j^{-p/2} \right)$ *is bounded by* $\mathrm{m}_{0/2}^{[\mathrm{K}\Delta t,1]}$.

Proof: To be able to compute the weights w_j and abscissas S_j of the quadrature (8.20), the fractional moments are transformed to integer ones, with non negative orders. Indeed, using the change of variable $r = \sqrt{S}$, $E_{p/2+n_q^-}$ can be written, for $k \in \{0, \dots, 2n_q - 1\}$:

$$E_{p/2+n_q^-} = \int_{\mathrm{K}\Delta t}^{1} s^{k/2-n_q^-} n(s)\mathrm{d}s - \sum_{j=1}^{n_q} w_j S_j^{k/2-n_q^-} = \int_{\sqrt{\mathrm{K}\Delta t}}^{1} r^k \mu(r)\mathrm{d}r - \sum_{j=1}^{n_q} w'_j r_j^k, \tag{8.22}$$

with $\mu(r) = (2rn(r^2))/(r^{2n_q^-})$ for $r \in [\sqrt{\mathrm{K}\Delta t}, 1]$, $r_j = \sqrt{S_j} \in [\sqrt{\mathrm{K}\Delta t}, 1]$ and $w'_j = w_j r_j^{-2n_q^-} \geq 0$. Then w'_j and r_j are defined as the unique weights and abscissas of the Gauss quadrature of order n_q corresponding to moments $m_k = \mathrm{m}_{k/2+n_q^-}^{[\mathrm{K}\Delta t,1]}$ of μ on $[\sqrt{\mathrm{K}\Delta t}, 1]$ for $k \in \{0, \dots, 2n_q - 1\}$.

By using this quadrature in equation (8.19), we cancel the $k + n_q^-$ first terms in the series, thus obtaining (8.21). The quantities $\bar{E}_{p/2}$ are bounded, since $S_j \geq \mathrm{K}\Delta t$ and

$$|\bar{E}_{p/2}| \leq \max\left\{ (\mathrm{K}\Delta t)^{p/2} \mathrm{m}_{-p/2}^{[\mathrm{K}\Delta t,1]}, (\mathrm{K}\Delta t)^{p/2} \sum_{j=1}^{n_q} w_j S_j^{-p/2} \right\} \leq \mathrm{m}_{0/2}^{[\mathrm{K}\Delta t,1]}.$$

□

However, it is still a rough bound, since this only leads to $\epsilon_{1/2} = O(\Delta t^{1/2})$, when Δt goes to zero, which is not sufficient to prove the convergence. An accurate estimation of the Gauss quadrature errors is needed to improve the bound, which should be investigated further in future works.

It seems now clear that the time evolution of the fractional moments depends on other moments than the ones involved in the moment system (4.24) and we cannot restrict the quadrature to the mere moments of positive order, which also corresponds to taking $n_q^- = 0$, if we want to be able to provide a good approximation of the moment evolution. This is thus the key issue with fractional moment evolution, since we need at least to consider some moments of negative order in the quadrature, i.e. $n_q^- > 0$. After a series of test-cases, we would be tempted to conjecture that for $n_q^- = 1$, where two supplementary moments of negative order ($m_{-1/2}^{[\mathrm{K}\Delta t,1]}$ and $m_{-2/2}^{[\mathrm{K}\Delta t,1]}$) are also represented by the quadrature, the solution approximates accurately the exact kinetic solution. We were not able to complete the proof of such a conjecture and some of these results are presented in Section 8.4. It seems that the terms $\bar{E}_{l/2}$ where $l = 1, 2$ in equation (8.21) cause the divergence of the standard algorithm proposed in section 8.2.3. This is an important result, since we need only a total of three quadrature nodes $n_q = 3$ to cancel the terms $E_{l/2}$ for $l = 1, 2$ and to capture correctly the kinetic evolution. Besides, we can use more quadrature points to increase the precision by choosing $n_q^- \geq 2$. Let us underline that the proper approximation and closure of the negative moments is here a key issue.

New adapted algorithm: according to the last results, we propose an adapted 4-steps algorithm. This algorithm is named **NEMO** (Negative Moments) algorithm and described below:

A. We reconstruct $n^{ME}(S)$ corresponding to the moment vector $\mathbf{m}(t_n)$ by the ME algorithm, then we calculate the disappearance flux as in (8.10),

B. We calculate the negative order moments in the interval $[\mathrm{K}\Delta t, 1]$

$$\mathrm{m}_{-a/2}^{[\mathrm{K}\Delta t,1]} = \int_{\mathrm{K}\Delta t}^{1} s^{-a/2} n^{ME}(s)\mathrm{d}s, \tag{8.23}$$

for $a = 1, \dots, 2n_q{}^-$, where $2n_q{}^- \geq 2$ is the number of additional moments of negative order used in this algorithm and chosen by the user. The other moments of positive order are computed using the disappearance flux:

$$\mathrm{m}_{k/2}^{[\mathrm{K}\Delta t,1]} = \mathrm{m}_{k/2}(0) - \Phi_{-,k/2}(t_n), \quad k = 0, \dots, 3, \tag{8.24}$$

where $\mathbf{\Phi}$ is the disappearance flux given in (8.10).

C. The abscissas $S_j = r_j^2 \in [\mathrm{K}\Delta t, 1]$ and the weights $w_j = w'_j r_j^{2n_q{}^-}$ of the quadrature corresponding to the moments $m_{p/2}^{[\mathrm{K}\Delta t,1]}$ for $p = -2n_q{}^-, \dots, 3$ are computed using the Product-Difference Algorithm Gordon(1968) corresponding to the moments $m_k = \mathrm{m}_{k/2+n_q{}^-}^{[\mathrm{K}\Delta t,1]}$ for $k \in \{0, \dots, 2n_q - 1\}$, leading to

$$m_{p/2}^{[\mathrm{K}\Delta t,1]} = \sum_{j=1}^{n_q} w_j S_j^{p/2}, \qquad p = -2n_q{}^-, \dots, 3 \tag{8.25}$$

D. Finally, we calculate the updated moments as follows:

$$\mathrm{m}_{k/2}(t_n + \Delta t) = \sum_{j=1}^{n_q} w_j (S_j - \mathrm{K}\Delta t)^{k/2}. \tag{8.26}$$

The singularities of the negative moments integrals, when Δt is very small, limits the use of high values of $n_q{}^-$. But in practice $\mathrm{K}\Delta t > 1.e-4$ and we will show that the choice of $n_q{}^- = 1$ or $n_q{}^- = 2$ are sufficient to obtain an accurate solution. In these cases, the integral computation of the negative order moments can be achieved correctly with 24 Gauss-Legendre quadrature points. For more complex evaporation laws, the algorithm can be straightforwardly generalized by computing the Lagrangian evolution of the abscissas. In other words, equation (8.26) becomes:

$$\mathrm{m}_{k/2}(t_n + \Delta t) = \sum_{j=1}^{n_q} w_j \widetilde{\phi}(t_n + \Delta t; t_n, S_j)^{k/2}, \tag{8.27}$$

and the integral of negative moments in (8.23) becomes:

$$\mathrm{m}_{-a/2}^{[\mathrm{K}\Delta t,1]} = \int_{\widetilde{\phi}(t_n;t_n+\Delta t,0)}^{1} s^{-a/2} n^{ME}(s) ds. \tag{8.28}$$

Remark 3 *The realizability of the moment is satisfied for this new algorithm. Indeed, the Product-Difference Algorithm ensures that the abscissas $S_j \geq \mathrm{K}\Delta t$ in equation (8.25). Therefore, the updated moments in (8.26) correspond to a positive distribution.*

8.3 Evaporation coupled with drag

In this paragraph, we present a coupled solver for the spray evolution under evaporation and drag force. The corresponding system of equations reads:

$$\left\{\begin{array}{rcl} d_t \mathrm{m}_0 & = & -\mathrm{K} n(S=0), \\ d_t \mathrm{m}_{1/2} & = & -\dfrac{\mathrm{K}}{2}\mathrm{m}_{-1/2}, \\ d_t \mathrm{m}_1 & = & -\mathrm{K}\mathrm{m}_0, \\ d_t \mathrm{m}_{3/2} & = & -\dfrac{3\mathrm{K}}{2}\mathrm{m}_{1/2}, \\ d_t\left(\mathrm{m}_1 \boldsymbol{u}\right) & = & -\mathrm{K}\mathrm{m}_0 \boldsymbol{u} + \mathrm{m}_0 \dfrac{\boldsymbol{u}_g - \boldsymbol{u}}{\theta}. \end{array}\right. \tag{8.29}$$

Since the first four equations do not depend on the last one, the updated moments can be computed using the **NEMO** algorithm. For the last equation, we use the method developed in Vié et al.(2013b) to solve the velocity evolution due to the drag force and evaporation.

The momentum evolution is conducted in two steps: first, we remove the part of the droplets, which will completely evaporate during the time interval $[t_n, t_{n+1}]$, by evaluating the disappearance fluxes of the moments and momentum:

$$\mathcal{U}^{[\mathrm{K}\Delta t,1]} = \mathcal{U} - \begin{pmatrix} \boldsymbol{\Phi}_- \\ \Phi_{-,2/2}\boldsymbol{u} \end{pmatrix}, \tag{8.30}$$

where $\mathcal{U} = (\mathrm{m}_{0/2}, \dots, \mathrm{m}_{3/2}, \mathrm{m}_{2/2}\boldsymbol{u}^t)^t$ and $\boldsymbol{\Phi}_-$ is the disappearance flux vector of the moments (8.10). Then, we use the lower principal representation to approximate the size distribution of the moments on the support $[\mathrm{K}\Delta t, 1]$ (8.25). The computation of the moments is achieved with (8.26). To update the momentum in this last step, we consider a size-velocity correlation, such that, at a given time $t \in [t_n, t_{n+1}]$, we attribute the velocity $\boldsymbol{c}_i(t)$ to each abscissa $S_i(t)$. Then, we write:

$$(\mathrm{m}_1 \boldsymbol{u})^{[\mathrm{K}\Delta t,1]}(t) = \sum_{i=1}^{n_q} w_i S_i(t) \boldsymbol{c}_i(t), \tag{8.31}$$

the abscissas $S_i(t)$ and the velocities $\boldsymbol{c}_i(t)$ are determined by solving the following ODE systems:

$$\left\{\begin{array}{rcll} \dfrac{d\boldsymbol{c}_i}{dt} & = & -\dfrac{\boldsymbol{u}_g - \boldsymbol{c}_i}{\theta S_i}, & \boldsymbol{c}_i(t=t_n) = \boldsymbol{u}(t_n), \\ \dfrac{dS_i}{dt} & = & -\mathrm{K}, & S_i(t=t_n) = S_i(t_n) \end{array}\right. \tag{8.32}$$

The final momentum is computed at $t = t_{n+1}$ using equation (8.31). The method can be generalized to more complex evaporation law by replacing $-\mathrm{K}$ in equation (8.32) by a general evaporation rate $\mathrm{R}_S(S)$, see appendix A.

8.4 Numerical results

This section is dedicated to some representative test-cases and analysis of the numerical results, to verify the robustness and the accuracy of the proposed numerical schemes. First, we test the new evaporation algorithm in the case of a d^2 law with $\mathrm{K} = 1$, for two different initial

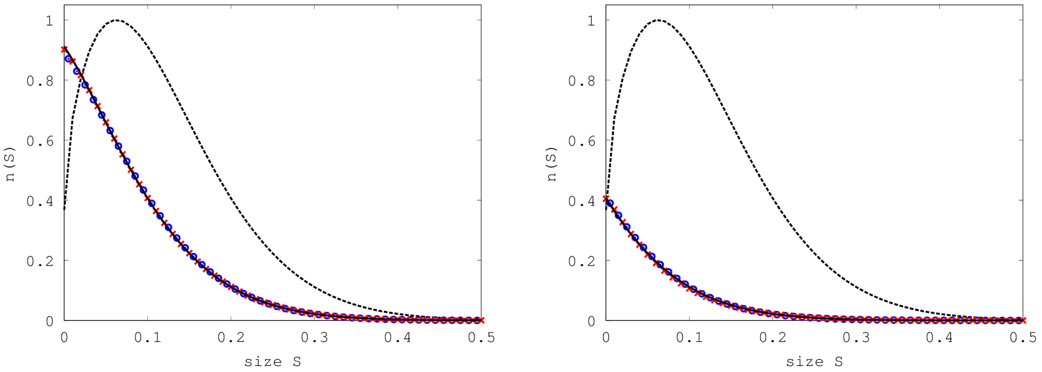


Figure 8.2: *Solutions of the reconstructed size distribution using:* ***NEMO*** *with* $n_q^- = 1$ *(cross), fully-kinetic (circle), exact kinetic solution (solid line) and the initial distribution (dashed line), at time* $t = 0.1$ *(left) and* $t = 0.2$ *(right).*

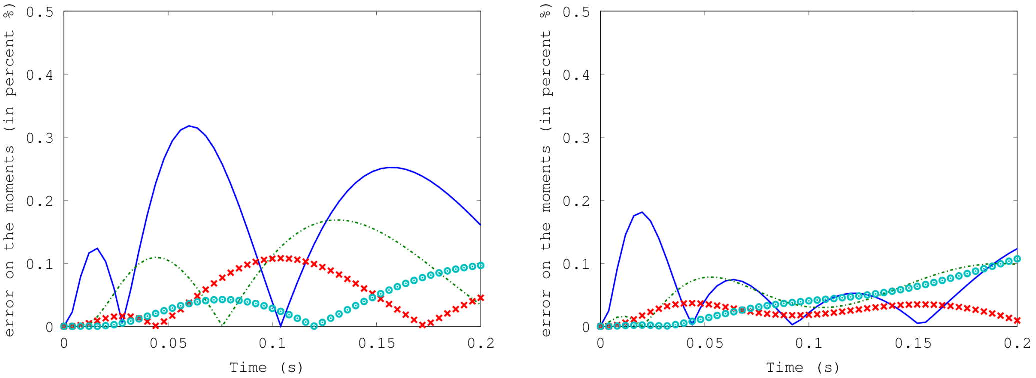


Figure 8.3: *Evolution of the relative moment errors (in percent):* m_0 *(solid line),* $\mathrm{m}_{1/2}$ *(dash-dotted line),* m_1 *(cross) and* $\mathrm{m}_{3/2}$ *(circle). Left: fully kinetic algorithm; Right:* ***NEMO*** *algorithm* $(n_q^- = 1)$*.*

conditions. An evaporation test in the case of non constant evaporation rate is also given as a complementary result, to show that the present algorithm is not restricted to d^2 law. Finally, a 2D case of an evaporating spray in the presence of a steady gas field, given by Taylor-Green vortices, is presented, in order to qualify the robustness and accuracy of the method compared to the EMSM in the same configuration. In this last simulation, we have used the second order Finite Volume Kinetic (FVK) scheme to solve the transport part. The same simulation using RKDG method is presented in chapter 10.

8.4.1 Evaporation in 0D simulation

8.4.1.1 Evaporation with a d^2 law for an initial smooth NDF

In this section, we use an initial smooth NDF, which is the same initial condition as the one used in Figure 8.1:

$$n^0(S) = exp(-20(S^{1/2} - 1/4)^2(S^{1/2} + 1)). \tag{8.33}$$

We underline that this distribution can be written in the form of a ME-reconstructed NDF, which is given in (4.37).

In this section, we compare the fully-kinetic and **NEMO** algorithms ($n_q{}^- = 1$), using the time step $\Delta t = 0.002$, with the exact kinetic solution. In Figure 8.2, we plot the ME-reconstructed NDFs from the solution of the moments obtained respectively by the two methods and the exact kinetic distribution at $t = 0.1$ and $t = 0.2$. One can see that the two ME-reconstructed distributions follow accurately the exact distribution. Capturing the correct size distribution using only four moments is not an obvious result. However, the results obtained with **NEMO** algorithm are more attractive, since we obtain the same accuracy using only three quadrature nodes compared with the 24-quadrature nodes of the fully kinetic scheme. To go further in the quantitative comparison, Figure 8.3 shows the evolution of the relative error of the four fractional moments. We can see that for the two algorithms the moment errors relatively to the initial values do not exceed 0.3%.

8.4.1.2 d^2 law evaporation for a discontinuous initial NDF

In this second case, we test the new algorithm **NEMO** in the case of a discontinuous initial NDF:

$$n^0(S) = \begin{cases} 1, & \text{if } S \in [0.1, 0.6] \\ 0, & \text{otherwise} \end{cases} \tag{8.34}$$

The initial NDF defined in (8.34) and the initial ME reconstructed NDF are plotted in Figure 8.4. In Figure 8.5, we present the reconstructed NDF computed using the the **NEMO** algorithm with $n_q{}^- = 1$ at two different times. In the same figures, we compare these results to the ones obtained with the fully kinetic solution and the exact kinetic solution[2]. As in the previous case, the **NEMO** algorithm shows an accurate prediction of the exact kinetic solution. Furthermore, Figure 8.6 shows that the relative error of the moments are less than 1.%, in the case of **NEMO** algorithm with $n_q{}^- = 1$ and $\Delta t = 6.e - 3$. This is an accurate result. However, we can see that the fully-kinetic algorithm is more accurate and the moment relative errors do not exceed 0.3%. We can improve the accuracy of the **NEMO** algorithm by using a smaller time step $\Delta t = 6.e - 4$ or by using $n_q{}^- = 2$ and keeping the same time step as the previous case. In Figure 8.7, we present the relative moment errors using **NEMO** algorithm with $n_q{}^- = 2$ and $\Delta t = 6.e - 3$. In this case, we are in the same level of accuracy as the fully kinetic algorithm.

8.4.2 Accuracy of the NEMO algorithm for a linear evaporation rate

NEMO scheme has been developed under the assumption of a d^2 law, but as it was explained before, the algorithm can be generalized to more complex laws by solving the Lagrangian equation (8.5) for each abscissas S_j given in the third step of the algorithm. In this section, we propose to evaluate the accuracy of the algorithm in the case where the evaporation rate depends linearly on the size:

$$R_s(S) = -(a + bS). \tag{8.35}$$

The exact kinetic solution can be computed according to equation (8.7). In the following, we set $a = 0.5$ and $b = 1$. Figure 8.8 presents the NDFs computed by the **NEMO** algorithm, the fully kinetic algorithm and compares them to the exact solution at $t = 0.3$ and $t = 0.6$. The relative error are given in Figure 8.9. We can see from these results, the accuracy of the generalized **NEMO** algorithm to predict the kinetic evolution.

---

[2] The exact kinetic solution is computed for the initial ME reconstructed size distribution.

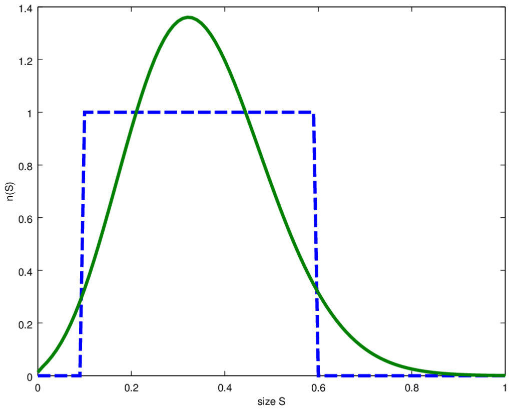


Figure 8.4: *The ME reconstructed NDF (solid line) and the initial discontinuous NDF (dashed line).*

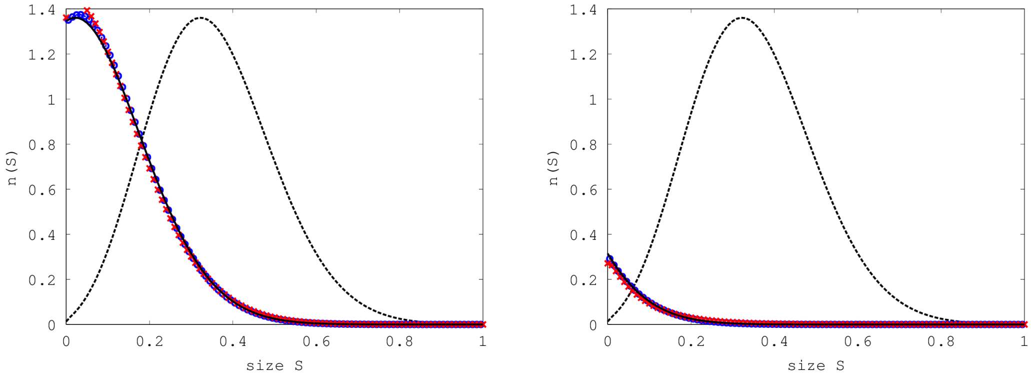


Figure 8.5: *Solutions of the reconstructed size distribution at at t = 0.3 (left) and t = 0.6 (right) using:* ***NEMO*** *with $n_q^- = 1$ (cross), fully-kinetic (circle), exact kinetic solution (solid line) and the initial distribution (dashed line).*

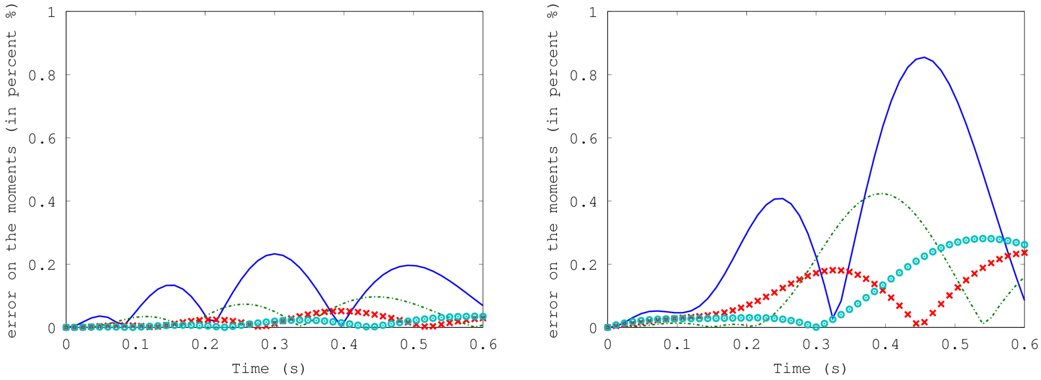


Figure 8.6: *Evolution of the relative moment errors (in percent):* m_0 *(solid line),* $\mathrm{m}_{1/2}$ *(dash-dotted line),* m_1 *(cross) and* $\mathrm{m}_{3/2}$ *(circle). Left: fully kinetic algorithm using $\Delta t = 6.e-3$; Right:* ***NEMO*** *algorithm using $n_q^- = 1$ and $\Delta t = 6.e-3$.*

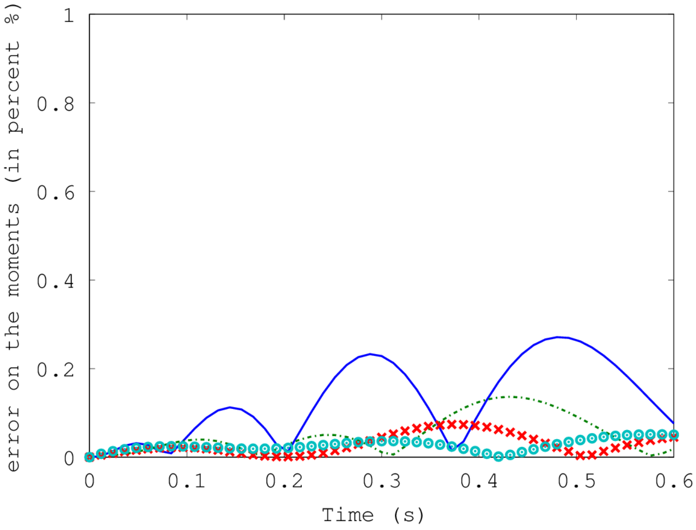


Figure 8.7: *Evolution of the relative moment errors (in percent) using* ***NEMO*** *algorithm* ($\boldsymbol{n_q}^{-}$ **= 2** *and* $dt = 6e-3$*):* m_0 *(solid line),* $\mathrm{m}_{1/2}$ *(Dash-dotted line),* m_1 *(cross) and* $\mathrm{m}_{3/2}$ *(circle).*

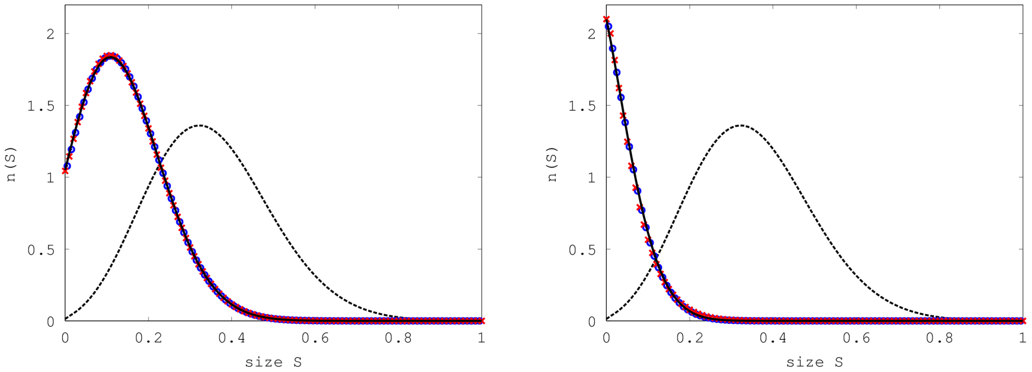


Figure 8.8: *The evolution of the NDF in the case of a linear evaporation rate: initial ME reconstructed solution (dashed line),* ***NEMO*** *algorithm using* $n_q^{-} = 1$ *(cross), fully kinetic algorithm (circle) and exact kinetic solution (solid line), at times* $t = 0.3$ *and* $t = 0.6$*.*

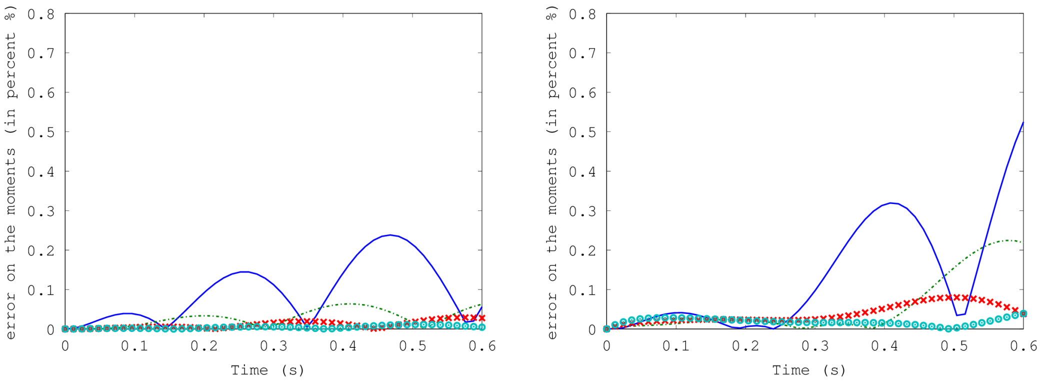


Figure 8.9: *Evolution of the moment errors (in percent) relatively to their initial value calculated with fully kinetic algorithm (left) and* ***NEMO*** *algorithm (right):* m_0 *(solid line),* $\mathrm{m}_{1/2}$ *(dash-dotted line),* m_1 *(cross) and* $\mathrm{m}_{3/2}$ *(circle).*

8.4.3 2D complete simulation: transport, evaporation and drag force

After the model and the numerical schemes have been tested in 0D and 1D, we propose in this part to compare in a classical 2D configuration, the present model and numerical schemes with the EMSM model, in a case where we consider transport, evaporation and drag. The simulations are performed using the CanoP code, developed within the collaboration of Maison de la Simulation, IFPEN and EM2C Laboratory. It is based on the `p4est` library Burstedde et al.(2011), a library providing highly scalable in massively parallel computations Adaptive Mesh Refinement (AMR) capability Drui et al.(2016a); Essadki et al.(2016). In the present simulation, we use only a uniform grid, since we are not concerned with AMR in the present work. In the following, we consider an evaporating spray in the presence of Taylor-Green vortices for the gas, which is a steady solution of the inviscid incompressible Euler equations. The non-dimensional velocity field of the gas is given as follows:

$$\begin{aligned} u_g(x,y) &= \sin(2\pi x)\cos(2\pi y), \\ v_g(x,y) &= -\cos(2\pi x)\sin(2\pi y), \end{aligned} \tag{8.36}$$

where $(x,y) \in [0,1]^2$ and we consider periodic boundary conditions. Initially, the spray is localized in the bottom-left vortex. The initial spatial size-distribution is given as follow:

$$n(t,\boldsymbol{x},S) = \mathbb{1}_{[a,b]}(S)\mathbb{1}_{\left\{x', ||x-x_c||_2<\sqrt{2}r\right\}}(\boldsymbol{x})\exp(-||\boldsymbol{x}-\boldsymbol{x}_c||_2^2/r^2), \tag{8.37}$$

where $[a,b] = [0.25, 0.75]$, $\boldsymbol{x}_c = (0.15, 0.15)$ and $r = 0.1$. The initial Stokes number computed according to the mean size $\bar{S} = \mathrm{m}_1/\mathrm{m}_0$ is equal to $\mathrm{St}(\bar{S}) = 0.05$, which is close to the critical Stokes number $\mathrm{St}_c = 1/8\pi \simeq 0.04$ de Chaisemartin(2009). The Stokes number decreases over the time because of the evaporation. The spray evaporation rate is $\mathrm{K} = 0.5$. Figures 8.10-8.11 present the computed spatial distribution of the volume fraction at two different times, using the EMSM model (left) and the new fractional moment model (right). We have used the second order KFV scheme for the transport resolution in both cases and the EMSM evaporation algorithm to solve evaporation in the EMSM model and the **NEMO** algorithm with $n_q{}^- = 1$ for the fractional moments. For the EMSM model, the volume fraction is not resolved but it is calculated through ME reconstruction of the size distribution $\alpha = (1/6\sqrt{\pi})\int_0^1 S^{3/2} n^{ME}(S)dS$. Instead, for the new model with fractional moments, the volume fraction is directly calculated by $(1/6\sqrt{\pi})\mathrm{m}_{3/2}$. The results of the two computations are closely similar, and the L^1-norm difference relatively to the initial volume fraction field is less than 3% at $t = 1$. This validates the results of the new model. In fact, the EMSM model was compared with the Multi-fluid model in Kah et al.(2012), and this comparison showed a high capacity of the EMSM model to predict the evaporation and the mean dynamics[3] of the spray.

---

[3] Both the EMSM model and the present model, with one size-section, are limited in predicting the size-conditioned dynamics compared with Multi-fluid model because the Stokes number is computed with mean size. The extension of the EMSM model to CSVM model Vié et al.(2013b) tackles these issues.

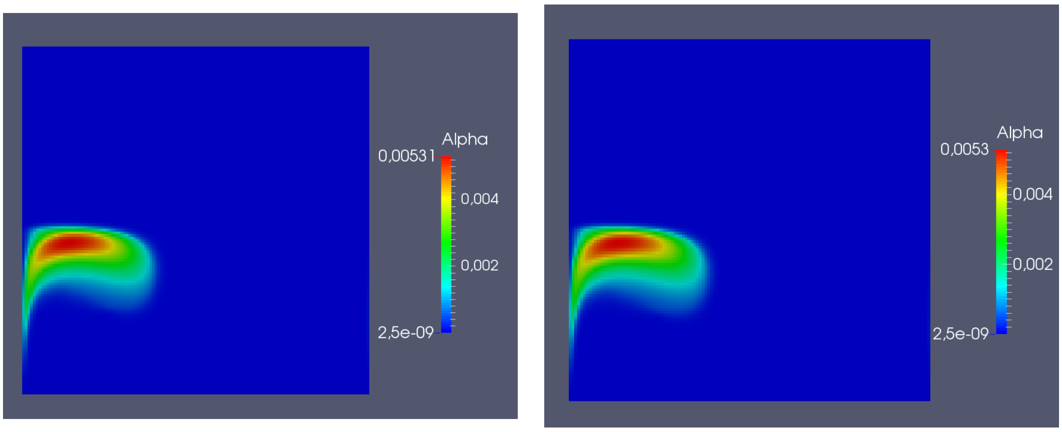


Figure 8.10: *The spatial distribution of the volume fraction for the Taylor-Green simulation at* $t = 0.5$. *The computation is carried out in a uniform grid* 128×128.

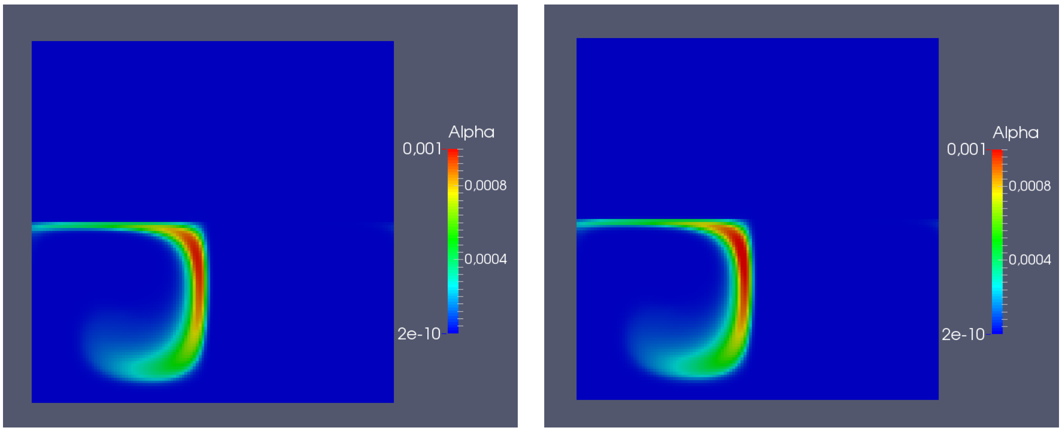


Figure 8.11: *The spatial distribution of the volume fraction for the Taylor-Green simulation at* $t = 1.0$. *The computation is carried out in a uniform grid* 128×128.

8.5 Conclusion

This chapter presents the time integration of the source terms involved in the fractional moments model. A new realizable algorithm to solve the evaporation is proposed in this work. The proposed strategy consists in making the link between the kinetic evolution of the NDF and thus capture the evolution of the fractional moments due to evaporation. We found that the moments of negative order should be taken into account for an accurate evaluation of the evaporation. The proposed **NEMO** algorithm, which is based on these developments, has improved remarkably the convergence and the accuracy of the solution. The new algorithm is validated in a set of 0D simulations. So far, the present chapter with chapters 4 and 7 give complete building blocks of a new spray model and dedicated robust and accurate numerical methods to simulate a polydisperse evaporating spray. A first result is presented in this chapter for a 2D academic configuration and compared to the EMSM model. In the next part of this manuscript, we show more advanced simulations using two different parallel codes, to evaluate the parallel performance.

Part IV

High performance computing and adaptive mesh refinement

Chapter 9

Implementing and testing the fractional moment model for spray simulation in an AMR code

9.1 Introduction

The two previous chapters were mainly dedicated to the numerical resolution of the system of equations of the fractional moments model to simulate an evaporating polydisperse spray. We showed in these chapters, how we can improve the accuracy of the resolution by using robust and accurate high order schemes. Nevertheless, in many computational fluid dynamics applications as in two-phase flow ones, the spatial representation of the phenomena is localized in thin regions. Therefore, very fine meshes may be needed in some specific locations of the computational domain to capture correctly these small scale structures, such as gas-liquid interface, flame front, etc. A uniform mesh would impose to use very fine meshes in the whole domain to obtain an accurate resolution. Consequently, the simulation would require high computational resources. When the small scales are localized in certain static zones of the domain, one may use fixed unstructured grids. In this case, the meshes are designed before the start of the simulation and kept fixed during the computation. The user then determines a priori the regions where high resolution is needed. However, for non-stationary and complex flows, such as the applications of our interest: an evolving gas-liquid interface in separated phases flow Drui(2017) and a dynamic spray in a combustion chamber, fixed unstructured grids fail to provide an accurate numerical resolution with low computational cost. Another alternative to reduce these computing requirements and to track a gas-liquid interface or dynamics of droplet segregations is to rather consider Adaptive Mesh Refinement (AMR) grids. Here, the meshes are refined or coarsened dynamically during the computation, thus, AMR can save computational time and memory in such applications, while ensuring an accurate numerical resolution.

However, developing an efficient AMR code can be a very complex task that needs to take into consideration different issues related to the management of adaptive meshes and parallel computations. Indeed, managing the meshes (refining/coarsening cells, iterating within the cells,...) is not free in terms of CPU and leads to an extra computational time (overhead of mesh management) compared to a pure Cartesian mesh. In order to have an interesting performance compared with standard uniform grids code, the overheads of AMR should be lower than the

CPU time spent in the numerical resolution. The second issue concerns the scalability of the AMR code in parallel computations. Refining and coarsening the meshes need to be accompanied with a load balancing algorithm to ensure the partition of the load between the different computational resources after each operation on the meshes. Finally, structured AMR, which is the considered method in the following, needs some special methods to represent complex domain geometries.

In this chapter, we present the `CanoP` code Drui et al.(2016a) that enables to solve systems of balance laws with finite volume schemes on 2D and 3D adptive grids. `CanoP` is based on the `p4est` library Burstedde et al.(2011). We present here the main features of this library, that allows to manage dynamically meshes in a massively parallel context. The implementation of the kinetic finite volume scheme in a non-conforming meshes, applied to the fractional moments model to simulate evaporating polydisperse spray, is also presented in this chapter. Finally, a series of test-cases is presented and analyzed, thus assessing the proposed approach and its parallel computational efficiency while evaluating its potential for complex applications.

9.2 AMR generalities

In the following, we consider only structured AMR (SAMR) methods. Contrary to unstructured AMR (UAMR), SAMR methods do not require to store explicitly all neighborhood relations (connectivities) between mesh elements. SAMR methods allow to develop an efficient mapping of the meshes, that can be encoded cheaply, compared to UAMR. However, SAMR methods are less flexible at mapping complex domain geometries compared to UAMR. This will require special treatments that we discuss in the present chapter. In the following, we simply substitute SAMR by AMR. In the literature, one finds two AMR approaches:

A. **Block-based** AMR methods Berger and Oliger(1984); Dubey et al.(2014) rely on pre-defined blocks (also called macro-meshes). Then, each block is refined or coarsened uniformly during the computation. Each macro-mesh consists of elementary meshes that can be mapped with regular grids and permit to reuse uniform mesh code. The Block-based AMR methods are easier to implement and do not need complicated encoding methods. However, they are less flexible in refining/coarsening the meshes since each macro-mesh has to be uniformly refined or coarsened, what is not optimal in terms of the compression rate (η) defined by:

$$\eta = \frac{N_{\text{unif}} - N_{\text{AMR}}}{N_{\text{unif}}}, \tag{9.1}$$

where N_{unif} is the number of cells in a uniform grid with spatial discretization as fine as the finest cells of the AMR grid, and N_{AMR} is the total number of cells in the AMR grid.

B. **Tree-based** or **cell-based** AMR methods are more flexible in refining and coarsening the meshes, which allows to reach higher compression rate. Indeed, the mesh refinement is performed through a recursive refinement of each cell independently of the other cells (see Figure 9.1). The recursive refinement stages of the mesh can be represented by a tree structure. The tree consists of nodes related by edges. Each node of the tree can be a parent of four (respectively eight) children in the case of 2D (respectively 3D) space domain or/and a child of another parent node. In this tree-structure, the nodes of the tree are called octant in 3D or quadrant in 2D. Each octant (respectively quadrant) is either a leaf

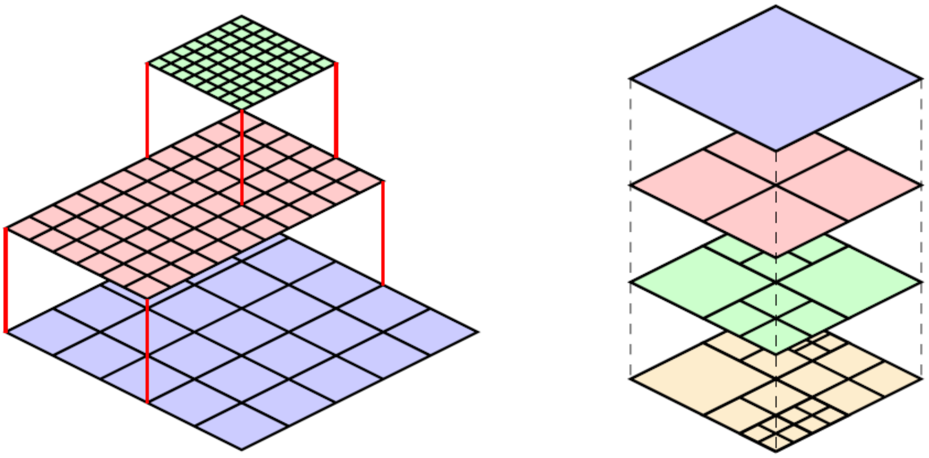


Figure 9.1: *Illustration of AMR techniques: left (block-based AMR method) and right (cell-based AMR method).*

or has eight (respectively four) children. The coarsen operation consists in replacing some leaves by their parent, while refining consists in creating new children quadrants/octants (new leaves) from a leaf quadrant/octant. The computational cells of the physical domain correspond to the leaves of the octree (in 3D) or quadtree (in 2D). It is then important to have an efficient encoding scheme. One could store the whole tree structure, since it contains all the connectivities between the cells. However, this solution stores unnecessary tree-data and consumes memory. Furthermore, the iteration among the cells (the leaves of the tree), like finding neighbors of a given cell, can lead in the worst cases to the whole tree traversal. For this reason, it is more convenient to consider a linear storage of the leaves using minimal possible mesh-data. It allows to encode cheaply the cell positions and connectivities in a linear array.

Some AMR definitions in this paragraph, we sum up some import definitions related to AMR and more specifically to the tree-based AMR approach, that will be used in this chapter.

- **Octree/quadtree**: these two words already used in the introduction of the tree-based approach, denote the recursive tree structure, where the nodes correspond to an octant (3D) or a quadrant (2D). An octree (respectively quadtree) can be associated with 3D (respectively 2D) domains.
- **level of refinement**: the level of refinement of an octant/quadrant corresponding to the tree root is zero. Recursively, the level of refinement of a child octant/quadrant l_c is its parent refinement level l_p plus one: $l_c = l_p + 1$.
- **non conforming mesh**: in AMR grids, a cell can have more than one neighbor at each of its faces. This occurs when the level of refinement is different between the two neighboring cells. We underline that, in such situation, one needs to adapt the numerical method to non-conforming meshes.
- **meta-data**: the set of data used to manage the mesh. They can contain the connectivity between a set of trees used to map a domain, cell coordinates, refinement level, etc.

From now on, by abuse of notation, we use the notation of 3D space domain(octant, octree, cube, etc.) even for 2D domains.

Different AMR libraries and codes are developed and referenced in the literature. We refer here to Drui(2017) for a general review of the existing AMR codes and methods. In the following, we choose to use the `p4est` library to manage adaptive refinement meshes in parallel context. The choice of this library is guided by its high parallel performance, it can allow an interesting compression ratio of the data by using the cell-based AMR approach, it is an agnostic library (independent of the numerical method and the models) and it can handle domains with complex geometries.

9.3 The `p4est` library

The `p4est` library is a tree-based and parallel AMR library written in C programming language. This library has shown a good parallel scalability up to $O(10^5)$ CPU cores Burstedde et al.(2011). The `p4est` library makes a heavy use of the z-order space filling curve to manage the meshes. It can map domains of complex geometry by using a set of octrees (called a forest). It also provides the user with different functionalities to manage the meshes in a parallel computation with sufficient flexibility to implement different numerical schemes and different models. In the following, we present the outlines of the `p4est` library: how the meshes are stored and managed and the main functionalities provided by this library to develop new applications.

9.3.1 Management of adaptive meshes

Octrees connectivity `p4est` enables to use an initial coarse-discretization of a physical space that can be defined manually by the user, or generated by some meshing software, such as `CUBIT` or `Gmsh`. The initial discretization, also called macro-mesh, allows to map complex domain geometry. However, it has to respect the following rules: 1-it is made of four-angled cells in 2D and hexahedral cells in 3D, 2-it should be **conforming**. In this representation, each macro cell covers a subdomain of the global domain that can be refined recursively into a tree structure (octree), where the collection of octrees that maps the whole domain is called here a forest of octrees. Each macro cell is mapped by a one-to-one correspondence to a reference cube. These macro-cubes represent the octant-root of the octrees. Their topological elements are: 6 faces in 3D and 4 in 2D; 12 edges in 3D (there is no edges in 2D); and 8 corners in 3D and 4 in 2D. Figure 9.2 illustrates the indexes of these elements used by `p4est`. The elements indexing depends on the orientation of the axes (x, y, z) used in the octree. Let us underline that, for two adjacent trees, `p4est` allows to have different coordinate systems. Therefore, the connectivity and orientation between the octrees are encoded in two arrays of dimension $K \times n_f$, where K is the number of the octrees, $n_f = 2d$ is the number of octree faces and d is the dimension of the physical space. The first array takes as entries an octree index $k \in \{0, 1, \ldots, K-1\}$ and a face $f \in \{0, \ldots, n_f - 1\}$ and returns the neighboring octree k' which shares the face f with the octree k: $NO(f, k) = k'$. For the same entries, the second array NF encodes in one integer number: the corresponding index number f' of the face f with respect to the octree k' and the relative orientation $r \in \{0, \ldots, 2^d - 1\}$ of the face f' with respect to the face f, since the orientation of the coordinate system can be different for two different octrees, such that we have $NF(k, f) = n_f r + f' \in \{0, \ldots, 2^d n_f - 1\}$. For more details about the connectivity encoding, we refer the reader to Burstedde et al.(2011).

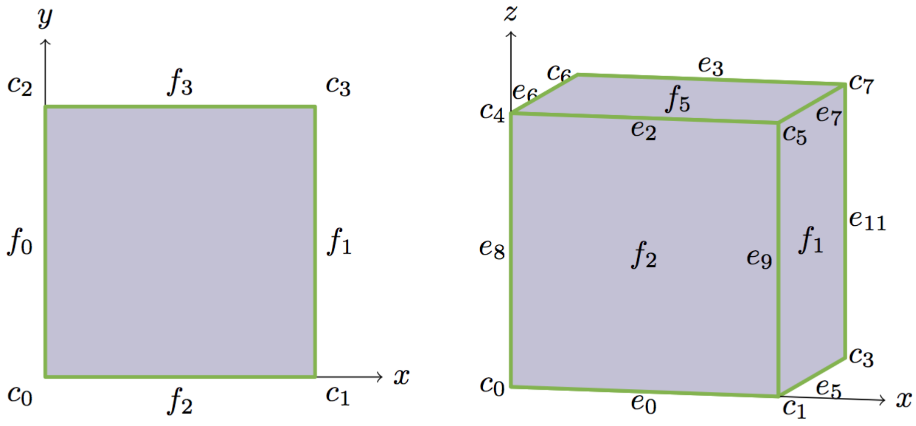


Figure 9.2: *Topological elements numbering. Reprinted from Burstedde et al.(2011).*

Linear storage and cells encoding in `p4est` , only the leaves of an octree are stored using the z-order curve (Morton's index). Since cells are linearly stored, its partition between MPI processes is simple. If the computational load is the same for all the domain cells, the target number of octants in the process p, denoted by N_p, can be derived from the total number of octants N and the number of processes P, according to Burstedde et al.(2011), as follows:

$$N_p = \left[\frac{N(p+1)}{P}\right] - \left[\frac{Np}{P}\right]. \tag{9.2}$$

In the case of non-uniform computational load between cells, the user can attribute a weight for each octant. The weight should be proportional to an estimated computational time in the octant.

The linear storage also simplify the iteration among the octant and the search of the neighbors. In forest of trees, an octant is uniquely defined by its corresponding octree number, the coordinate position of its front lower left corner and the level of refinement. First, `p4est` groups the octants per-octree array. Therefore, It does not require to store the octree number associated to each octant. For each octree, `p4est` stores the position $x, y, z \in \{0, \dots, 2^b - 1\}$ of the lower left corner of an octant respect to the coordinate system of the octree and the level of refinement $0 \leq l \leq b$, where b is the maximum level of refinement fixed in the `p4est` library. The relative coordinates of an octant in an octree are stored with $d \times b$ bits. The integer-based representation of the octant coordinate positions avoids topological errors due to roundoff of floating-point errors. Finally, to store an octant position in a linear array, `p4est` encodes the position (x, y, z) in an integer m using $d \times b$ bits as follows:

$$\overline{m}^2_{3i+2} = \overline{z}^2_i, \quad \overline{m}^2_{3i+1} = \overline{y}^2_i, \quad \overline{m}^2_{3i} = \overline{x}^2_i, \tag{9.3}$$

where $0 \leq i \leq b - 1$. The notation $\overline{\bullet}^2$ indicates numbers written in base 2 and $\overline{\bullet}^2_i$ is the i-th bit of the binary representation. We underline that the physical coordinates are computed through the transformation mapping function between the macro-mesh and its cube $[0, 2^b - 1]^d$ octree-representation.

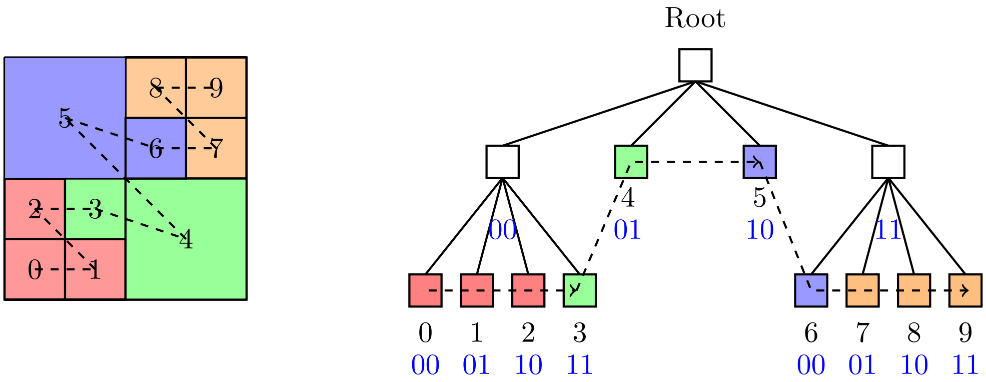


Figure 9.3: *z-order traversal of the quadrants in one tree of the forest and load partition into four processes. Dashed line: z-order curve. Quadrant label: z-order index. Color: MPI process.*

9.3.2 Main functionalities of `p4est`

In this section, we summarize the main functionalities of `p4est` used to manage the meshes in parallel simulations. The library `p4est` has been developed to serve as a parallel cell-based AMR-grid generator and manager for an increasingly large spectrum of physical applications, since there is a decoupling between the mesh and the numerical resolution. Moreover, due to its flexibility, `p4est` library is easy to use in the development of new numerical simulation codes. In fact, while `p4est` handles the meta-data of the mesh structure, it is up to the user to define the application data per cells such as: density, velocity, energy, moments, etc, as well as the functions that can be used in the numerical computation such as: update the data, compute the reconstruction variables (slopes, gradients, interpolations, etc.) used mainly for high order numerical schemes, mark the cells to be refined or coarsened, etc. The functions defined by the user is used as callback functions of `p4est` functions. Here, we summarize the main `p4est` functions as follows:

- Create new refined forest and repartition the load between MPI processes.
- Iterate among quadrants and call a call-back function defined by the user to mark the quadrants to be refined (creation of new children) or coarsened (remove the quadrants from the tree).
- Ensure 2:1 balance (the size ratio between two neighbouring quadrant does not exceed two).
- Partition the linearly stored mesh between the processes after each modification of the trees.
- Communicate the data of the quadrants located at the boundaries of each "processor-decomposition" domain to the neighboring processors, by mean of ghost quadrants.

To sum up, the p4est library only manages the meta-data associated with the mesh structure, and no application is associated with the library. The user has to specify the application data, also called user-data in this chapter, and implements the functions that will perform tasks in the user-data. These user-defined functions are mainly call-back functions called by the `p4est` functions defined above.

9.4 The `CanoP` code

`CanoP` is a C/C++ code, that can be used to solve systems of balance laws with Finite Volume (FV) schemes on 2D and 3D adaptive grids. The management of the meshes is based on the `p4est` library, where `p4est` functions are used to create the meshes, allocate the memory, refine and coarsen meshes and distribute the load between MPI processes. `CanoP` can be considered as an abstraction layer of `p4est` library, which facilitates the implementation of new applications and benefits from a flexible use of `p4est`. Thus, the implementation of new applications can be achieved with minimum programming efforts: the user needs to specify a data model specific to his application (for example: density, velocity, energy, etc.) encapsulated in a C++ object, which is denoted by *user_data* and he needs to define specific functions (function to compute numerical fluxes of a FV scheme, update the solution, mark cells to be refined or coarsened, etc.). These functions defined by the user are used as callback functions of other common `CanoP` functions or `p4est` functions.

`CanoP` code is developed within a collaboration between **La Maison de la Simulation**, **EM2C** laboratory, **CMAP** laboratory and **IFPEN**. The collaboration between these institutes has allowed to implement different physical applications using different FV schemes:

- a scalar advection upwind solver was implemented during CEMERACS 2014 summer school, and it was the first application in `CanoP`. The objective, during this project, was to experience the `p4est` library and its performances in parallel computations Drui et al.(2016a),
- Euler's equations with gravity source terms are solved using a HLLC scheme for the convective part and an elliptic solver for the gravity source term. This model is used mainly for astrophysics applications,
- 3-equation system to model two-phase flows is solved numerically with a Suliciu's solver Drui et al.(2016b). This model is tested and compared with some experimental data and DNS code results Drui(2017),
- spray models based on moment methods (fractional and integer size moments), presented in chapters 2 and 3, are also implemented and tested in `CanoP` code Essadki et al.(2016),
- and a new two-phase flow model, coupling a moment model for spray with a two-fluid model for separated phases is still under development.

`CanoP` code is permanently evolving, because of the continuous integration of new applications. Indeed, the main objective of this code is to facilitate for new users the integration of new applications without modifying the main code structure. Therefore, an effort has been recently conducted to split the code into two parts:

- **The solver part** is the `CanoP` core, common to the different applications and the different libraries used in this code. It is in this part where `CanoP` calls the different `p4est` functions to manage the mesh and defines some common functions for all `CanoP` applications.
- **The application part** is dedicated to the user, where he defines his desired data models *user_data* and specific functions to be used as callback functions called by `p4est` function or other common functions defined in the solver part.

In order to simplify even more the process of integrating new application, a refactoring of the code using C++ classes and templates has recently been conducted to make the solver part more generic. New applications can now be implemented without the need to modify in the solver

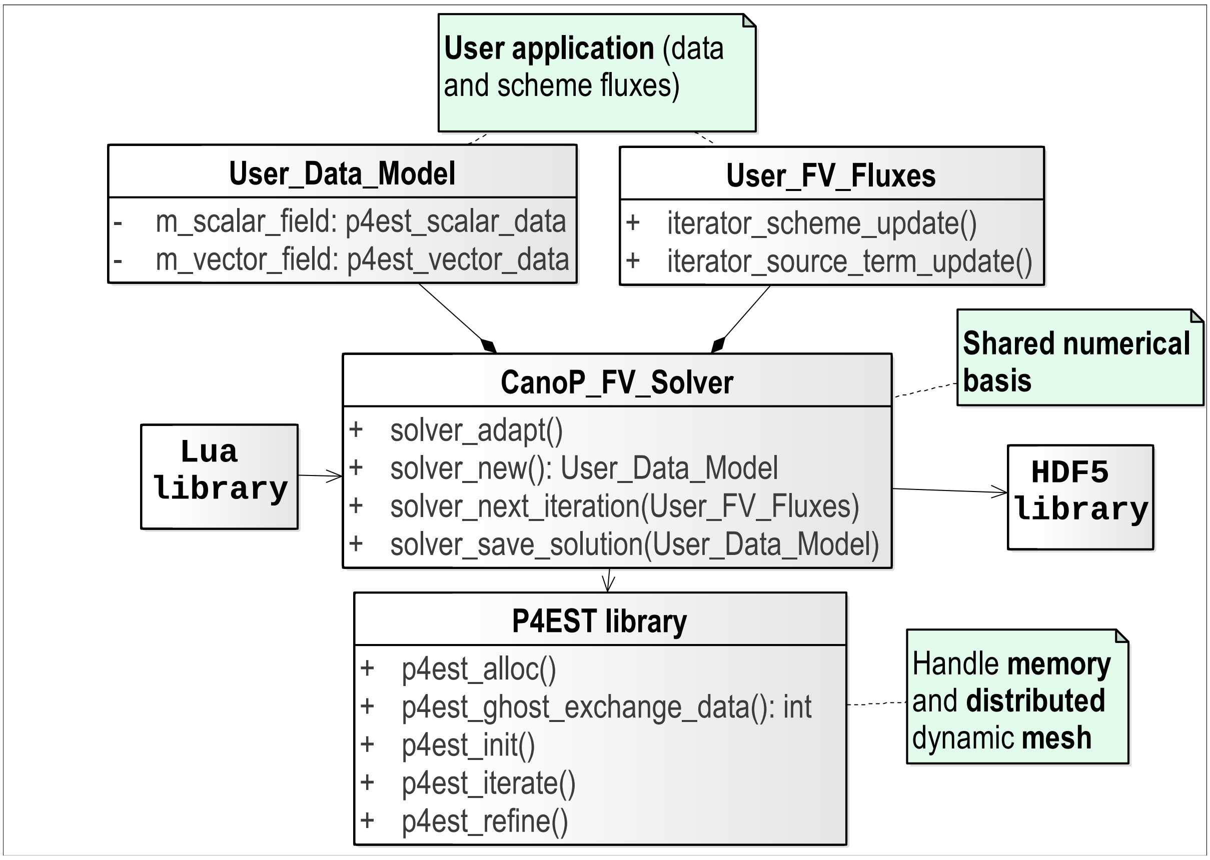


Figure 9.4: *General `CanoP` architecture.*

part.

9.4.1 `CanoP` code architecture

In this section, the main features and the structure of the code are presented. Figure 9.4 illustrates the main code architecture and its links with the different libraries. This figure shows the articulation between the application part, where specific functionalities are implemented and the solver part which contains common functionalities and interfaces between the application part and different libraries:

- The HDF5 library used for parallel output,
- The LUA library used to read input data,
- The `p4est` library.

The main stages of the code are summarized in Figure 9.5. In these steps, we first have the initialization step, which consists in initializing the MPI processes, reading the LUA input file and instantiating C++ classes according to the setting parameters of the input file. Then, in the time loop step, we update the solution, adapt the meshes in each iteration and write the solution in parallel manner using HDF5 library functions. Finally, in the finalization step, all created objects and pointers are destroyed.

The different steps are detailed further in Figures 9.6 to 9.9. For mesh initialization, we have

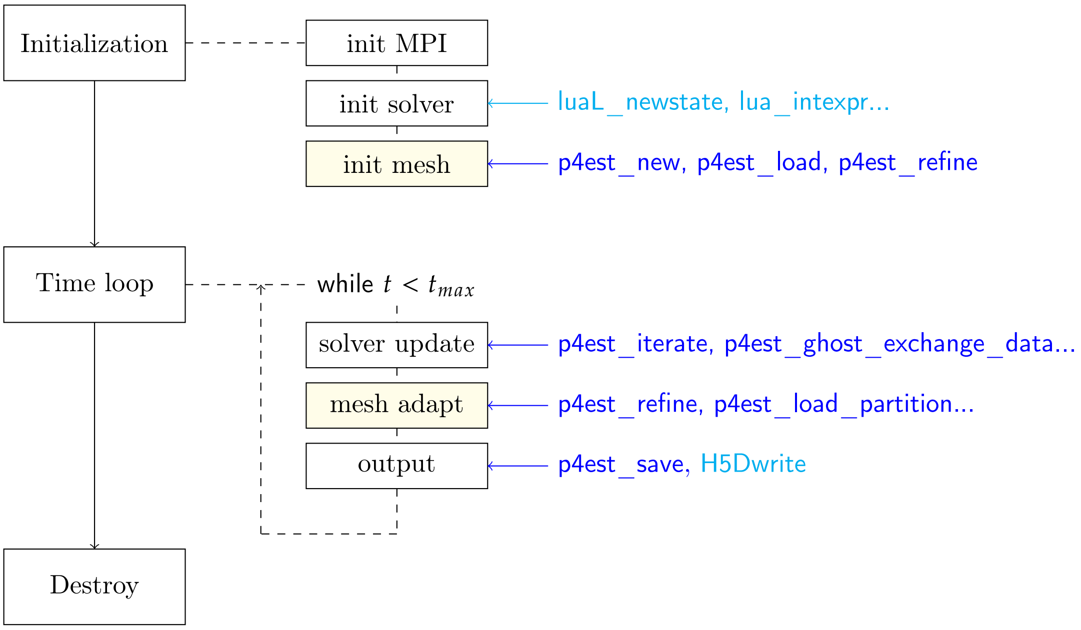


Figure 9.5: *Sketch of the `CanoP` code structure and calls for `p4est` functions, reprinted from Drui(2017).*

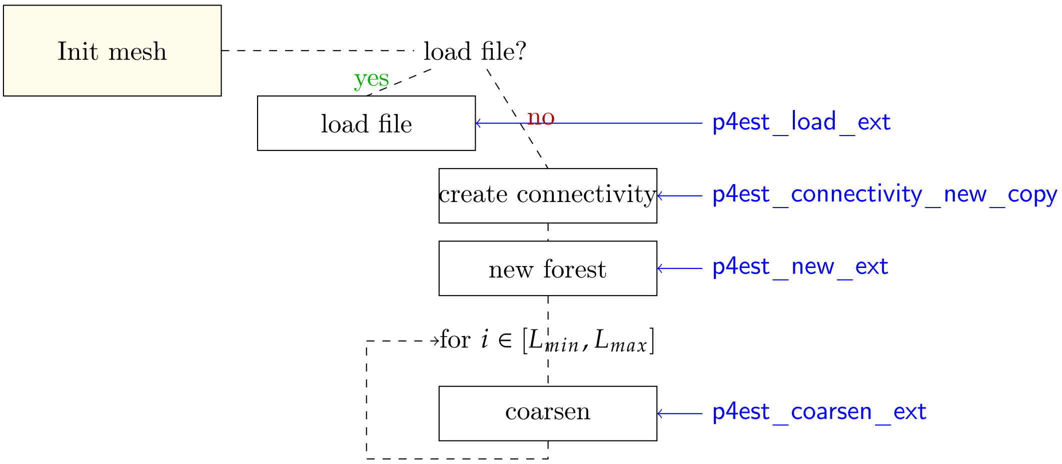


Figure 9.6: *Zoom in the init part structure and calls for `p4est` functions, reprinted from Drui(2017).*

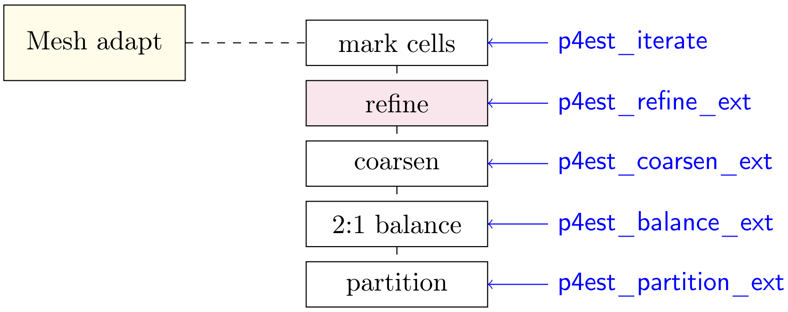


Figure 9.7: *Zoom in the mesh adapt part structure and calls for `p4est` functions, reprinted from Drui(2017).*

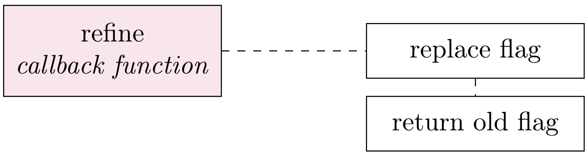


Figure 9.8: *Zoom in the refine callback function, that informs `p4est` if the cell should be refined. Reprinted from Drui(2017).*

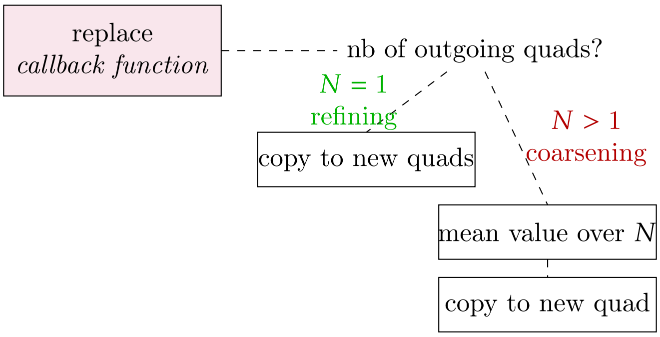


Figure 9.9: *Zoom in the replace callback function, that computes the value of the newly created quadrants. Reprinted from Drui(2017).*

two possibilities: the first consists in loading the meta-data (mesh data) and solution data from a previous output simulation or to create new mesh and a new initial solution. For mesh adaptation (Figure 9.7), the function *p4est_iterate* is called to iterate over all the octants and flag cells to be coarsened or refined according to the refinement criterion and the refinement thresholds that are provided by the parameters of the simulation. Then, *p4est_refine_ext* and *p4est_coarsen_ext* are called to refine or coarsen the meshes. In this second stage, two types of data have to be updated: the meta-data (`p4est` mesh data) and the solution data (*user_data*) in the new cells. To refine and coarsen the meshes, `CanoP` uses the flags determined in the previous step and calls the replace callback function to copy the data of a parent octant in its children octant when it is about refining a cell or average the old children variables in their new parent octant when it consists in coarsening cells. Once the adaptation of the mesh is over, the function *p4est_partition_ext* is called to redistribute the load between the processors.

9.4.2 Object-oriented programming in `CanoP`

The first versions of `CanoP` were written in a standard C language. Consequently, the implementation of new application has often required to modify the code structure and to use preprocessor conditions (#ifdef, etc) each time some parts of the code were not generic. Difficulties have then arisen about the code readability and maintenance. A recent refactoring of the code and using object-oriented programming (OOP) C++ language allowed to improve its readability and the extensibility. Indeed, the actual version benefits from the different advantages of C++ classes, template classes and inheritance. The re-factored code proposes:

- a clear modular structure of the code through abstract classes,
- a complete separation between the solver part and the application part allowing new users to implement new applications without affecting the rest of the code,
- a simplify maintenance and a better readability.

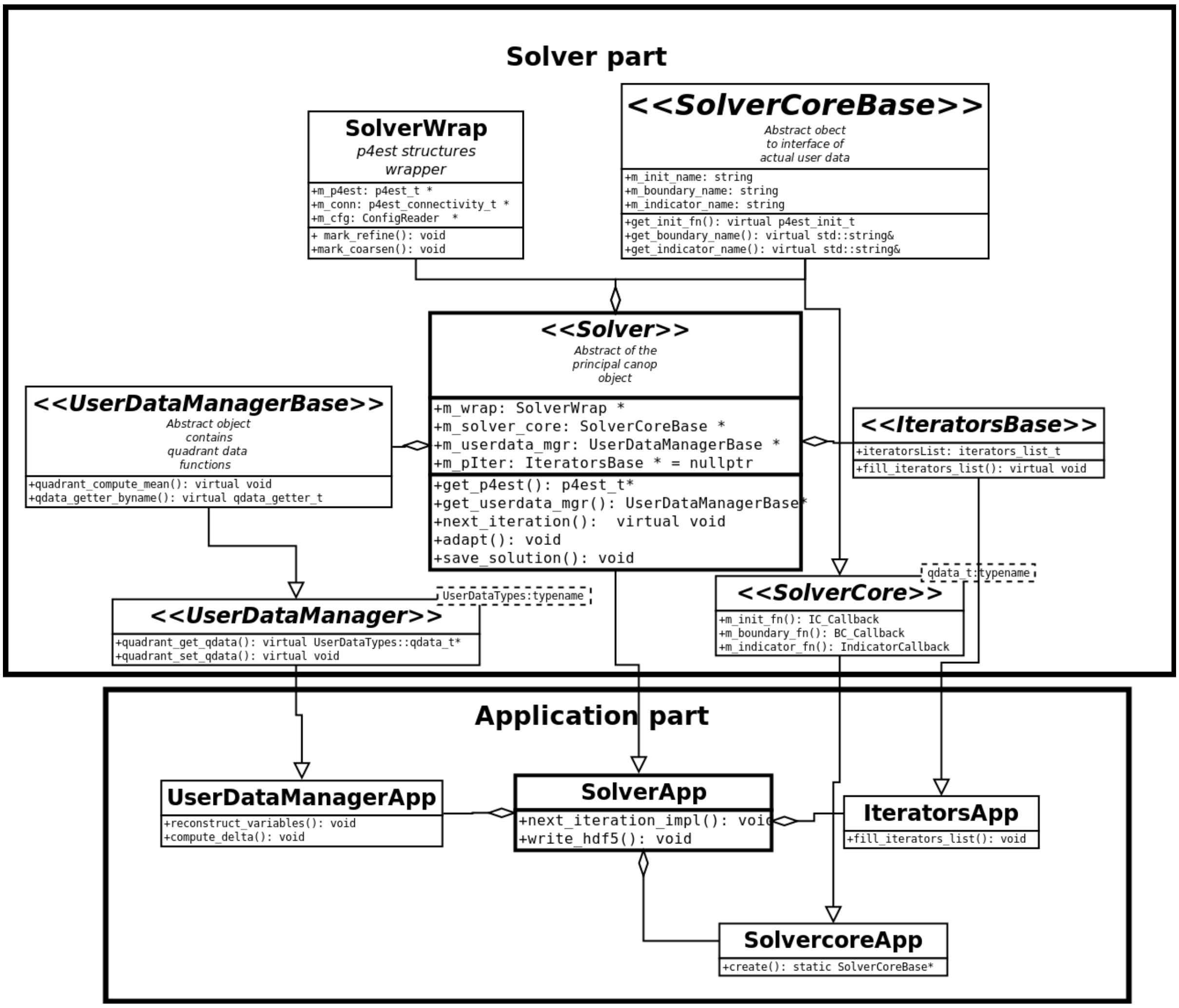


Figure 9.10: *Diagram of the principal class objects in `CanoP` code.*

Figure 9.10 illustrates the main classes used in the code, which are separated into abstract classes that belong to the solver part and implemented classes defined in the application part. The classes in the application part inherit from the abstract classes of the solver part. The class *Solver* is the principal class that provides all necessary information of the simulation: meta-data (`p4est` mesh data), user data (data per cell) and simulation parameters. The parent class contains:

- the class *SolverWrap* encapsulating the `p4est` meta-data,
- the class *SolverCoreBase* defining an interface to the *user_ data*,
- the class *UserDataMangerCore* containing quadrant data functions,
- and the class *IteratorsBase* defining the iterators functions (the callback function called by *p4est_ iterate*).

All these classes are abstract classes and are instantiated in the initialization step (see the previous section), where constructors of the inherited classes of the application part are used in this step. We underline that each application defines its owns classes derived mainly from the abstract classes of the solver part and Figure 9.10 illustrates only a case with one application.

To sum up, `CanoP` was developed with the idea to simplify the use of `p4est` library and to share it for several applications more easily. Now, `CanoP` provides to new users a simple framework to implement their applications and benefit from the high performance of the `p4est` library. A new implementation will only consist in providing the data model (*user_ data*), implementing the numerical scheme (a function to compute the numerical flux for the finite volume scheme) and the refinement criterion or using the already existing implementations.

9.5 Numerical resolution of a spray model in a cell-based AMR

9.5.1 The kinetic finite volume scheme on non-conforming meshes

The kinetic finite volume (KFV) numerical scheme presented in section 7.2 dedicated to the resolution of the transport part of the system of moment equations (7.1), needs to be adapted to non conforming meshes. As in chapter 7, we use a dimensional splitting. In the following, we consider only the transport step in the x-direction. Let us denote by q a 2D non conforming cell verifying the $2:1$ size balance, which means that the size ratio of two neighboring cells is either 1, 2 or 1/2. In general, we write the finite volume scheme in q as follows:

$$\boldsymbol{U}_q^{n+1} = \boldsymbol{U}_q^n - \frac{\Delta t}{\Delta x_q}(\boldsymbol{F}_{q+1/2} - \boldsymbol{F}_{q-1/2}), \tag{9.4}$$

where $\boldsymbol{U} = (\mathrm{m}_{0/2}, \dots, \mathrm{m}_{3/2}, \mathrm{m}_{2/2}u, \mathrm{m}_{2/2}v)^t$ is the vector of conservative variables of the system (7.1) and $q-1/2$ (resp. $q+1/2$) refers to the left (resp. right) faces in the x-direction. To illustrate how we adapt the scheme in a 2:1 balanced mesh, let us consider the case when the cell q has two neighboring cells on his right: η_1^r and η_2^r as is illustrated in Figure 9.11. The flux across the common edge depends on the states of the three cells q, η_1^r and η_2^r:

$$\boldsymbol{F}_{i+1/2} = \boldsymbol{F}(\boldsymbol{U}_q^n, \boldsymbol{U}_{\eta_1^r}^n, \boldsymbol{U}_{\eta_2^r}^n), \tag{9.5}$$

The exact expression of the fluxes, derived from the integral expression of the moments and the kinetic solution (7.13), is:

$$\boldsymbol{F}_{q+1/2} = \boldsymbol{F}_{q+1/2}^{+} + \boldsymbol{F}_{q+1/2}^{-}, \tag{9.6}$$

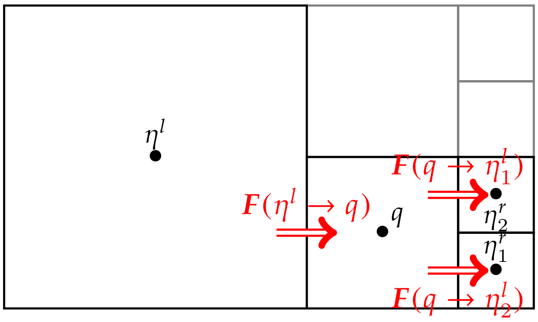


Figure 9.11: *Non-conforming grid and the different fluxes associated with cell q in one space direction.*

such that

$$\boldsymbol{F}^{+}_{q+1/2} = \frac{1}{\Delta y_q \Delta t}\int_{\boldsymbol{x}=(x,y)\in q} \boldsymbol{U}(t_n,\boldsymbol{x})\mathbb{1}_{\Sigma_q^{+}(\Delta t)}(x,y)dxdy, \tag{9.7}$$

and

$$\begin{aligned}\boldsymbol{F}^{-}_{q+1/2} &= \frac{1}{\Delta y_q \Delta t}\Bigg(\int_{\boldsymbol{x}=(x,y)\in \eta_1^r} \boldsymbol{U}(t_n,\boldsymbol{x})\mathbb{1}_{\Sigma^{-}_{\eta_1^r}(\Delta t)}(x,y)dxdy \\ &\quad + \int_{\boldsymbol{x}=(x,y)\in \eta_2^r} \boldsymbol{U}(t_n,\boldsymbol{x})\mathbb{1}_{\Sigma^{-}_{\eta_2^r}(\Delta t)}(x,y)dxdy\Bigg).\end{aligned} \tag{9.8}$$

Where for a cell c, $\Sigma_c^{\pm} = \{(x',y) \in c, \pm(x_{c+1/2} - \Delta t u(t_n,x')) < \pm x'\}$ is the set of droplets in the cell c that will reach its right (resp. left) side during the time $[t_n, t_{n+1} = t_n + \Delta t]$.

In the case of first order scheme, the solution is constant per cell and the numerical flux can be expressed as follows:

$$\boldsymbol{F}_{q+1/2} = \frac{\Delta x_{\eta_1^r}}{\Delta x_q}\boldsymbol{F}\left(\boldsymbol{U}_q, \boldsymbol{U}_{\eta_1^r}\right) + \frac{\Delta x_{\eta_2^r}}{\Delta x_q}\boldsymbol{F}\left(\boldsymbol{U}_q, \boldsymbol{U}_{\eta_2^r}\right), \tag{9.9}$$

where the flux $\boldsymbol{F}\left(\boldsymbol{U}_i, \boldsymbol{U}_j\right)$ is computed as in the case of a uniform mesh, using equation (7.15). The variable Δx_i is the length of the cell i.

For the second order, we use affine reconstruction of the transported variables: canonical moments and velocity as explained in chapter 7. However, in the case of non-conforming meshes, there are different ways to reconstruct the solution based on the states of the neighboring cells and their positions. A first possibility consists in estimating a gradient tensor of the transported variables from the whole neighboring cells. In `CanoP`, we use a simple way to compute the slopes. We illustrate the method in the case of Figure 9.11: cell q has a double-size neighbor η^l on its left and two half-size neighbors η_1^r and η_2^r on its right. We first compute the slopes of the reconstructed variables (canonical moments and the velocity) as in (7.21) using the cells (η^l, q, η_1^r), then we compute the slopes in the same way but using now (η^l, q, η_2^r). The final slopes are computed as the minmod limiter of the two last slopes. Since cells of different sizes are used and the limitation of the slopes is rather restrictive, second order is not guaranteed near extrema of the solution. However the robustness and the realizability of the method are ensured. Moreover the results show a good accuracy improvement compared to the first order scheme.

9.5.2 Refinement criterion

The determination of an efficient criterion remains a complex subject, since the simulation depends on the physical phenomena and on the numerical methods. Multiresolution analysis provides some robust tools to compress the data using wavelet functions and control the compression errors at each refinement level (see Muller(2003), Harten(1995) and Duarte(2011) and the references therein). In `CanoP`, these methods are difficult to implement because the tree structure does not respect some necessary properties (like threaded tree, see Harten(1995)) for multiresolution methods. Therefore, we use a heuristic criterion based on some consideration of

the main results cited in Harten(1995).
Harten Harten(1995) introduced a method to control the error, based on an interpolating function to approximate the variables at refined level $l+1$ from the coarsen level l, then he computes the details which are the difference between the predictive variables and the true variables. The details are considered as the errors at level l. Following Harten's idea, we implement a refinement criterion adapted to `CanoP` for spray applications. The method consists in estimating the moment m_0 for the children quadrants of a given parent quadrant q. The prediction is based on the slope computation in each direction for each face depending on the neighbor quadrants not necessary of the same size as the quadrant q[1].

$$\tilde{m}_0(x,y) = \begin{cases} m_{0,q} + Dm_0^l(x-x_q) + Dm_0^u(y-y_q), & \text{if } (x,y) \in child_1 \\ m_{0,q} + Dm_0^r(x-x_q) + Dm_0^u(y-y_q), & \text{if } (x,y) \in child_2 \\ m_{0,q} + Dm_0^l(x-x_q) + Dm_0^d(y-y_q), & \text{if } (x,y) \in child_3 \\ m_{0,q} + Dm_0^r(x-x_q) + Dm_0^d(y-y_q), & \text{if } (x,y) \in child_4 \end{cases} \tag{9.10}$$

The subscripts l, r, d, u refer respectively to the left, right, down and up edge. Dm_0^j is the slope variation in the direction x if $j = l$ or $j = r$ and y otherwise. Finally, the children index follows the z-order.

The slopes are computed from the neighboring quadrants. To illustrate this calculation for the slopes in direction x, we consider the case of Figure 9.11:

$$\begin{aligned} Dm_0^l &= \frac{m_{0,q} - m_{0,\eta^l}}{(\Delta x_q + \Delta x_{\eta^l})/2}, \\ Dm_0^r &= \frac{(m_{0,\eta_1^r} + m_{0,\eta_2^r})/2 - m_{0,q}}{(\Delta x_q + \Delta x_{\eta_1^r})/2}. \end{aligned} \tag{9.11}$$

Let us emphasize that the reconstruction in (9.10) does not necessarily respect some physical and mathematical properties (the conservation of the variables, the maximum principle and the realizability). In fact, the reconstruction is only used to estimate the error and is not used in the numerical scheme.
Then we compute the L1-norm of the difference between $m_{0,q}$ and the predicted $\tilde{m}_0(x,y)$.

$$err(m_0)_q = \int_q |m_{0,q} - \tilde{m}_0(x,y)| dxdy, \tag{9.12}$$

where $|q|$ is the total surface of the quadrant q.

This estimated error is used as a refinement criterion, such that the error is compared to a given threshold ξ and if $err(m_0)_q > \xi$ then the quadrant is refined. If all siblings of a given quadrant verify $err(m_0)_q < c\xi$ then the quadrant is marked for coarsening. The parameter $c \leq 1$ is chosen by the user.

9.6 Results

This section is dedicated to some numerical results using `CanoP` code. We use the KFV numerical scheme to solve the convective part of the system of equations (7.1). This scheme is presented

---

[1] For Harten's predictor operator, the author used neighbors of the same level to predict the variable in the next high level).

in chapter 7 for the case of uniform meshes. In the present chapter, we use its extension in the case of non-conforming meshes. The resolution of the source term is detailed in chapter 8. The robustness and the accuracy of these methods have been verified in the last two chapters in a set of 0D, 1D and 2D uniform Cartesian tests. Now, we aim at proving that 1- the AMR solver reaches a very nice level of accuracy once we have chosen a proper refinement criterion, 2- the proposed strategy is valid for complex droplet models, that is for source term with a much higher level of complexity leading to much higher arithmetic intensity and 3- that we have a high level of scalability and efficiency of the parallel implementation of the numerical strategy relying on realizable schemes and splitting methods.

To do so, we have chosen three test-cases:

- The first case consists in a localized spray in the presence of Taylor-Green (TG) vortices, Figure 9.12. This case though simple, does illustrate a realistic configuration occurring in automotive engine, where the droplets are concentrated in some local regions, while large vacuum regions can be found everywhere else. This first case is a non evaporating spray. Here, we only assess the performance of the AMR calculation compared to a uniform grid and then study the scalability at a given level of accuracy.
- The second case is the same as the first one, but with an evaporating spray. The objective of this simulation is twofold. First, we show the robustness and the accuracy of the evaporation algorithm coupled with transport and drag. Second, the evaporation algorithm is representative of the high arithmetic intensity within an embarrassingly parallel configuration using operator splitting. The aim is to illustrate the impact of the AMR techniques coupled to the operator splitting on the computational cost for intensive source terms, such as the inversion algorithm used to reconstruct the approximated spray density function.
- The third case is a non evaporating Homogeneous Isotropic Turbulence in 2D and 3D. A comparison of droplet segregation is conducted for an AMR Eulerian solution with a uniform Eulerian solution and a Lagrangian reference solution. This comparison shows the efficiency of the AMR Eulerian solution to predict correctly the main physical features with low computational resources. The gain in the CPU time is obtained in this framework of a much richer configuration in terms of physics, while in a really non-favorable case since the spray is present in the whole domain.

For the three cases, we fix the CFL (7.25) at $CFL = 0.9$ and the constant c used in the refinement criterion at $c = 2^{-(d+1)}$, where d is the space dimension .

9.6.1 Droplet cloud in Taylor-Green vortices

We simulate a non evaporating polydisperse spray in a two dimensional periodic domain in the presence of Taylor-Green vortices for the gas. This gas velocity profile is a steady solution of the inviscid incompressible Euler equations. The non-dimensional velocity field of the gas is given by the following expression:

$$\begin{aligned} u_g(x,y) &= sin(2\pi x)cos(2\pi y), \\ v_g(x,y) &= -cos(2\pi x)sin(2\pi y), \end{aligned} \tag{9.13}$$

where $(x,y) \in [0,1]^2$.
Initially, we consider a motionless cloud located in the bottom-left vortex in Figure 9.12.

The initial spatial distribution of the spray is Gaussian inside a small disk of a radius $r = 0.1$ and equals to zero outside. The initial size distribution, uniform inside the disk, is equal to 1

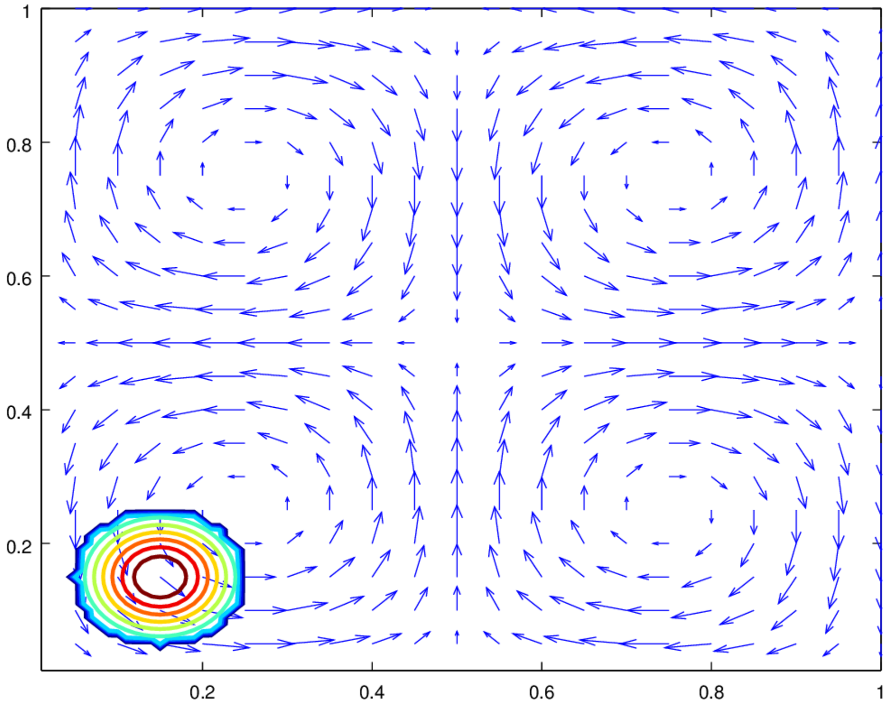


Figure 9.12: *Stationary gaseous velocity vector field of the Taylor-Green vortices and spray initial number density.*

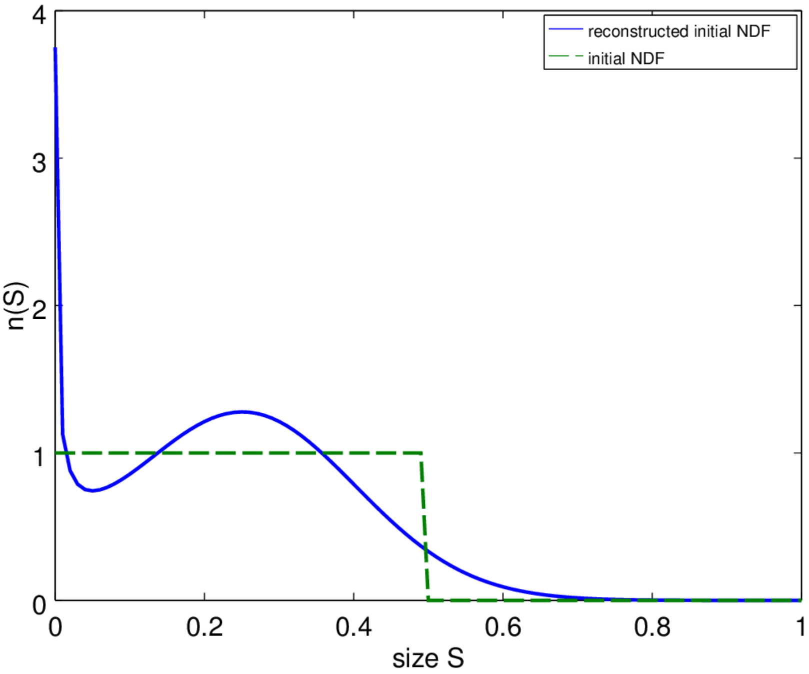


Figure 9.13: *The initial NDF in the dashed line and its reconstruction through ME in the solid line.*

for $S \in [0, S_{mid}]$ and 0 otherwise. The moments are thus given by:

$$m_{k/2}(t=0,\boldsymbol{x}) = \frac{2}{k+2}(S_{mid}^{(k+2)/2})exp(-||\boldsymbol{x}-\boldsymbol{x}_c||_2^2/r^2), \tag{9.14}$$

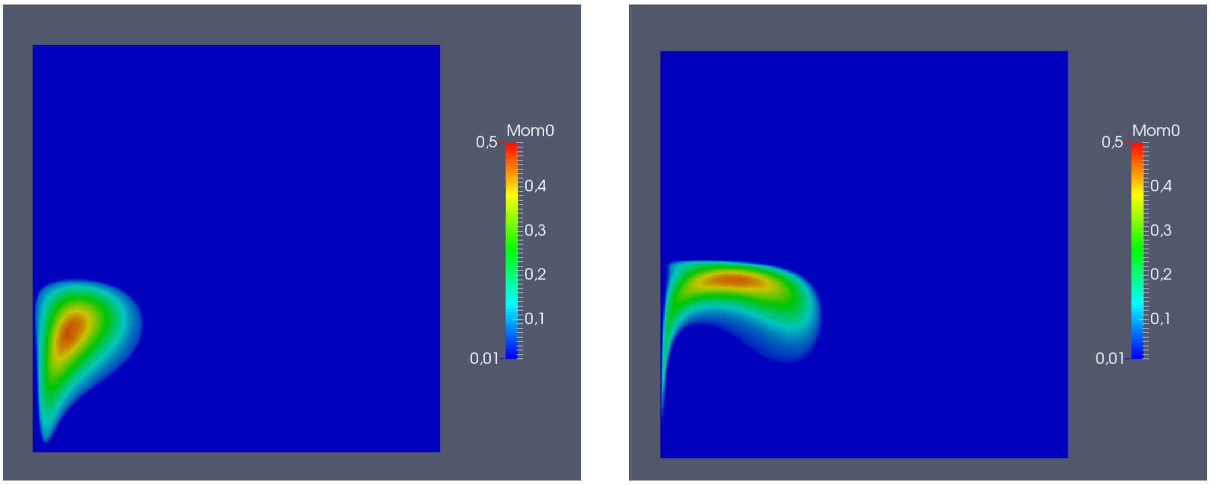
where $k = 0\ldots 3$, $S_{mid} = 0.5$ and $||\boldsymbol{x}-\boldsymbol{x}_c||_2^2 \leq r$. The initial exact size distribution as well as its Maximum Entropy reconstructed density $n_{ME}(S)$ from the four fraction moments (9.14) are presented in Figure. 9.13 for $\boldsymbol{x} = \boldsymbol{x}_c$.

We consider low inertia droplets such that the mean Stokes number[2] $\mathrm{St} = 0.025$ is less than the critical value $\mathrm{St}_c = 1/(8\pi)$. It has been shown in de Chaisemartin et al de Chaisemartin(2009), that for $\mathrm{St} < \mathrm{St}_c$ the droplets stay in their origin vortex and cannot travel from one to another, while, for $\mathrm{St} \geq \mathrm{St}_c$ they are ejected out of their original vortices. In this last configuration, we encounter particle trajectory crossings, that are not taken into account in the monokinetic assumption considered here and lead to delta-shocks. Notice that these singularities are completely handled numerically, thanks to the Kinetic Finite Volume (KFV) numerical scheme and the robustness is ensured as it was shown in chapter 7.

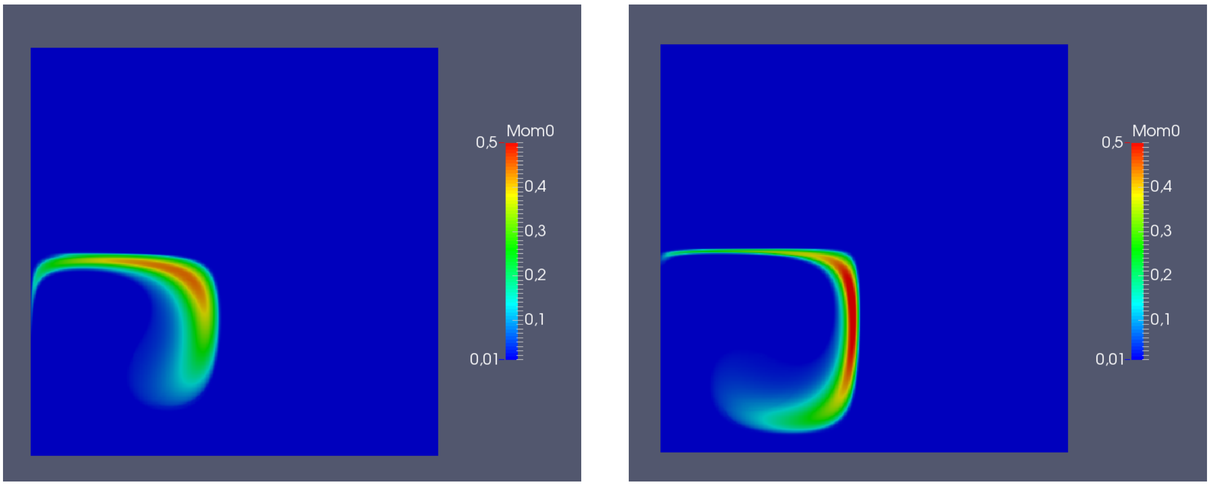
In Figure 9.14, we display the spatial evolution of the number density m_0 using a second order scheme and AMR grid at different times, as well as the mesh grid given at two instants in Figure 9.15. The maximum refinement level is $l_{max} = 9$ and the minimum level is $l_{min} = 4$. We use the refinement criterion presented in the section 9.5.2 and choose a small threshold $\xi = 1.e-7$.

---

[2] The mean Stokes number is computed with respect to the mean droplet size $\overline{S} = \mathrm{m}_{2/2}/\mathrm{m}_{0/2}$.



$t = 0.25$ $t = 0.5$



$t = 0.75$ $t = 1.$

Figure 9.14: *Taylor-Green simulation using the second order scheme in adaptive refinement grid, the maximum level is* $l_{max} = 9$ *and the minimum level is* $l_{min} = 4$

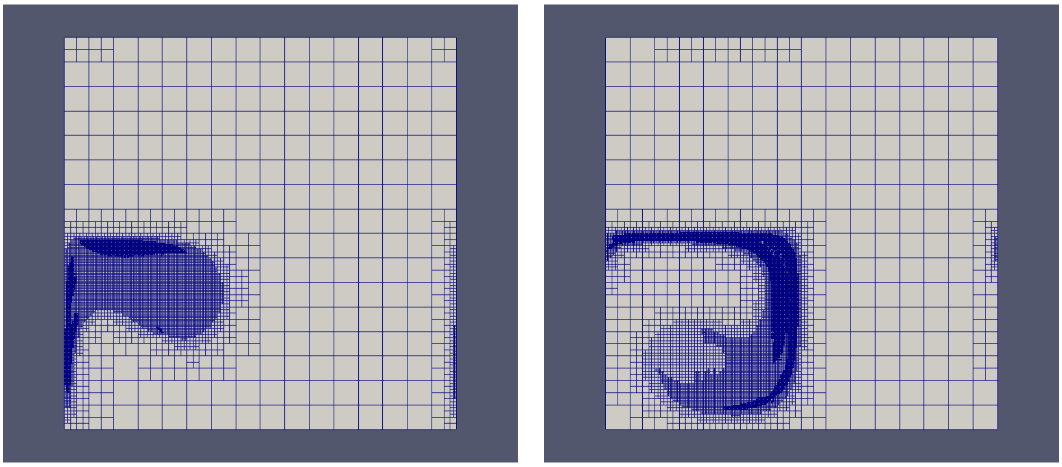


Figure 9.15: *The AMR grid in the case of Taylor-Green with non evaporating spray, with* $l_{max} = 9$*,* $l_{min} = 4$ *and threshold* $\xi = 5.e - 7$*: at* $t = 0.5$ *(left) and* $t = 1.0$ *(right).*

9.6.1.1 Refinement criterion and solution quality

In this section, we study the influence of the refinement criterion in the accuracy of the solution and its impact on the computational time. The objective is to show that we can keep an accurate solution while reaching a high compression rate, thus saving time and memory. It also allows to set up a fair framework in order to conduct a parallel efficiency study in the next subsection. After investigating the influence of the refinement criterion depending on the order of the scheme, we focus on the intrinsic accuracy of the solution.

Threshold ξ	relative L1-error	compression rate %	CPU time
$1.e - 7$	0.7%	87.3%	5.81
$5.e - 7$	5.2%	92.4%	3.89
$1.e - 6$	10.1%	94.4%	3.24
$5.e - 6$	29.6%	97.3%	1.71
$1.e - 5$	35.4%	97.8%	1.15

Table 9.1: *Relative* L^1*-error, compression rate and CPU time depending on the threshold* ξ *using the first order scheme.*

In Tables 9.1 and 9.2, we present the results for the first order and the second order scheme using 36 processors. The L^1-error is computed relatively to the solution on a refined uniform grid 512×512, considered as the reference solution. The compression rate and the CPU time are also provided. The purpose is to evaluate the impact of the compression rate on both accuracy and computational efficiency.

The two tables show a significant CPU time saving compared to the reference solution. First for the first order scheme, the CPU time of the reference solution (uniform meshes) is $29.15s$. Using the smallest threshold $\xi = 1.e - 7$, we reduce the computation time by a factor of five with a good level of precision. We could save more CPU time and memory (see the compression rate) by increasing the threshold but we lose the accuracy of the solution. For $\xi \geq 1.e - 6$, the relative L^1-error is more than 10%. Since the computing time of the second order finite volume scheme is much larger than the communication and mesh adaptation time, we save more CPU time in this case, whereas the CPU time used in the computation of the second order reference

Threshold ξ	relative L1-error	compression rate %	CPU time
$1.e-7$	0.6%	88.8%	28.38
$5.e-7$	3.2%	92.6%	19.32
$1.e-6$	5.3%	94.7%	14.6
$5.e-6$	15.6%	97.2%	6.9
$1.e-5$	19.7%	97.7%	3.73

Table 9.2: *Relative L^1-error, compression rate and CPU time depending on the threshold ξ using the second order scheme.*

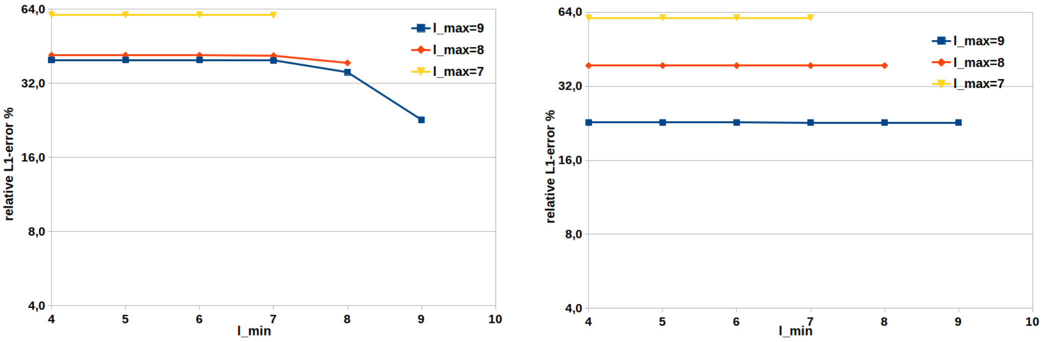


Figure 9.16: *L^1-error for the first order scheme in logarithm scale versus the minimum level of compression l_{min} plotted for different maximum refined levels l_{max}: using $\xi = 1.e-6$ (left) and $\xi = 1.e-7$ (right).*

solution is much larger: $231.63s$. By using mesh adaptation with the threshold $\xi = 1.e-7$, we reduce this CPU time by a factor of eight, maintaining a good precision. We could save more CPU time and memory by choosing $\xi = 1.e-6$ with a good accuracy (relative L1-error $\sim 5\%$). Let us underline that these two tables do not compare the solution accuracy depending on the scheme order because the relative L^1-error is computed for each order as the difference between the solution on a uniformly refined grid and the solution for an adaptively refined grid using the same order.

In a second study, we consider a common reference solution for the first and second order simulations. The reference solution is computed using a second order scheme on a uniform grid of 1024×1024 cells, which corresponds to the level $l = 10$. The two Figures 9.16 and 9.17 show that the relative L^1-error does not evolve much when we vary the minimum level except when we reach a uniform grid. However, for a given maximum level of refinement, there is a strong impact of ξ on the ability of the adapted solution ($l_{min} < l_{max}$) to capture an accurate solution. In fact for each maximal level of refinement, there is a threshold in terms of refinement criterion above which compressing the solution deteriorates its quality. Its is clear in Figure 9.16 and 9.17 that $\xi = 1.e-7$ is below this threshold and $\xi = 1.e-6$ is above it. For $\xi = 1.e-7$ the relative-error remains almost constant and we can use mesh refinement while preserving an accurate solution, whereas for $\xi = 1.e-6$, we can observe that mesh refinement for $l_{max} = 9$ results in a clear increase of the error. Once we have found a well chosen threshold, for which the accuracy of the uniform and AMR solutions are equivalent, we can then study the computational efficiency and parallel performance of the numerical strategy.

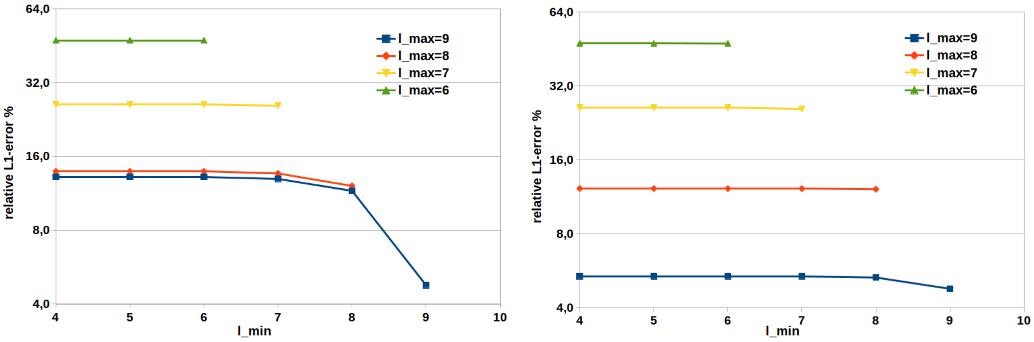


Figure 9.17: *L^1-error for second order scheme in logarithm scale versus the minimum level of compression l_{min} plotted for different maximum refined levels l_{max}: using $\xi = 1.e-6$ (left) and $\xi = 1.e-7$ (right).*

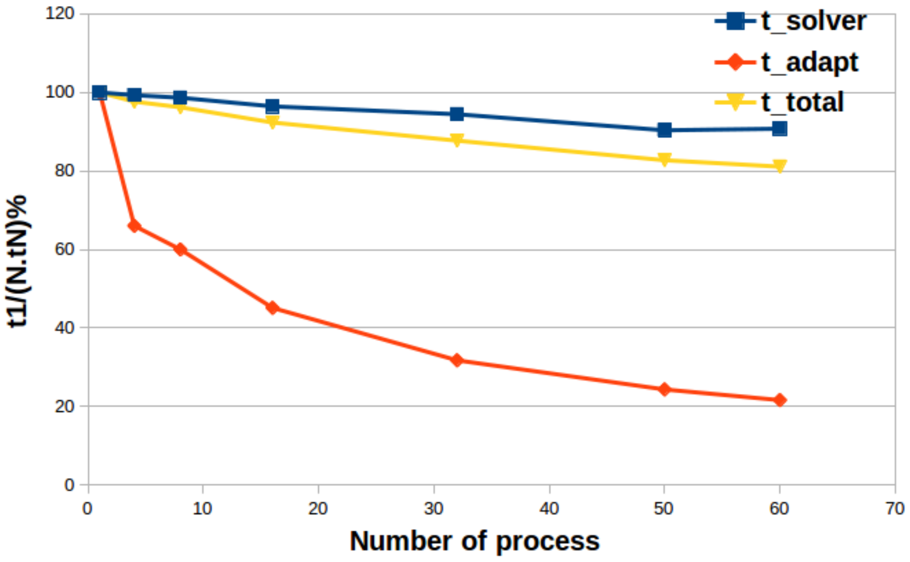


Figure 9.18: *Strong scaling of the second order scheme in AMR grid, the maximum level is $l_{max} = 9$ and the minimum level is $l_{min} = 4$.*

9.6.1.2 Parallel performance

We assess the parallel performance by measuring the computational time of various operations:

- Solver time corresponds to the time needed in the finite volume scheme and source evaluation in the splitting strategy,
- Adaptation time corresponds to the time of refinement, coarsening, partitioning and 2:1 size balancing,
- And finally the total time which is the sum of both solver and adaptation times without I/O.

We use the second order Taylor-Green simulation in an adaptive refinement grid such that the maximum level $l = 9$ corresponds to the finest meshes and the minimum level $l = 4$ corresponds to the coarsest meshes. We choose the previously obtained threshold $\xi = 5.e-7$ to ensure the quality of the compressed solution.

In Figure 9.18, we display strong scaling results. The simulation is performed in CentraleSuplec cluster (2 Intel Xeon procs, memory per node 24Go, total core number 156) using up to 60 MPI processes. The solver time (transport and drag) shows an efficiency of more than 88%. The adaptation time shows a weak efficiency due to the large time spent in the communication between processors essentially for the balance algorithm Burstedde et al.(2011). Nevertheless, the global time of computation shows a good parallel performance and the efficiency reaches 81% for 60 processors. In fact, the solver time represents 85% of the total CPU time. Let us underline that the number of effective cells in this simulation with a compression rate of about 90% is close to 30,000 so that an efficiency of 81% on 60 cores is very satisfactory.

9.6.2 Taylor-Green evaporating spray

Using exactly the same configuration with Stokes number St = 0.025, we switch to an evaporating spray with K = 1/2. Since we use an inversion algorithm in order to reconstruct the NDF from the moments by ME algorithm in each cell and time step for the evaluation of the disappearance flux of droplet at zero size, we expect a significant computational cost related to source terms. Thus, this simulation is representative of the complexity that we will have to face by switching to more complex droplet models in realistic configurations.

Table 9.3 presents the computational time ratio spent in solving each part of the numerical resolution: the transport and the source terms, using two uniform grids (256×256 and 512×512) and running the simulation on 96 processors. As expected, the resolution of the source terms requires around 95% of the total time. It is then interesting to investigate how the mesh adaptation behaves in such a configuration, where the operator splitting technique, used here to solve independently the source terms in various cells, can lead to an embarrassingly parallel configuration.

	Transport solver	Source term solver
512×512	2.2%	97.5%
256×256	5.6%	93.5%

Table 9.3: *Time ratio (in percent) spent in transport solver and source solver (including evaporation and drag) using* 96 *MPI processes on a uniform mesh.*

In the previous study without evaporation, $\xi = 5.e - 7$ was shown to be an adequate threshold for a high compression and good solution quality, therefore we maintain this value in the present case. Figure 9.20 shows the spatial number density at four different times of the evaporating Taylor-Green simulation on AMR grid ($l_{max} = 9$ and $l_{min} = 4$) and Figure 9.21 presents the corresponding meshes. Compared to the simulation running on the uniform grid 512×512, the relative L^1-error of the AMR solution computed at $t = 0.5$ is 3.9%, the compression rate is 97.7% and we have reduced the computational time by a factor of more than 45. In fact, the CPU time used in the resolution of evaporation is much larger than the CPU time that we spend in the resolution of the transport and the mesh adaptation. More precisely, in this simulation, the CPU time spent in the evaporation resolution is $76.5s$, the transport resolution $2.4s$ and the mesh adaptation $5.2s$, which means that the evaporation alone takes more than 90% of the total CPU time, Table 9.4 gives further details of the time ratio spending at each operation using

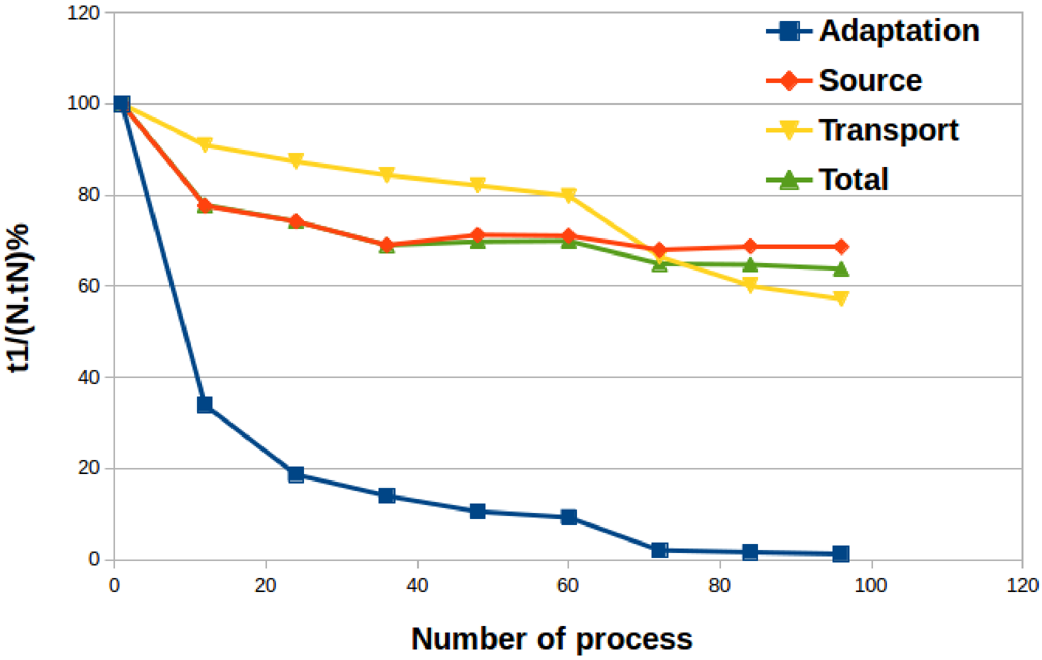
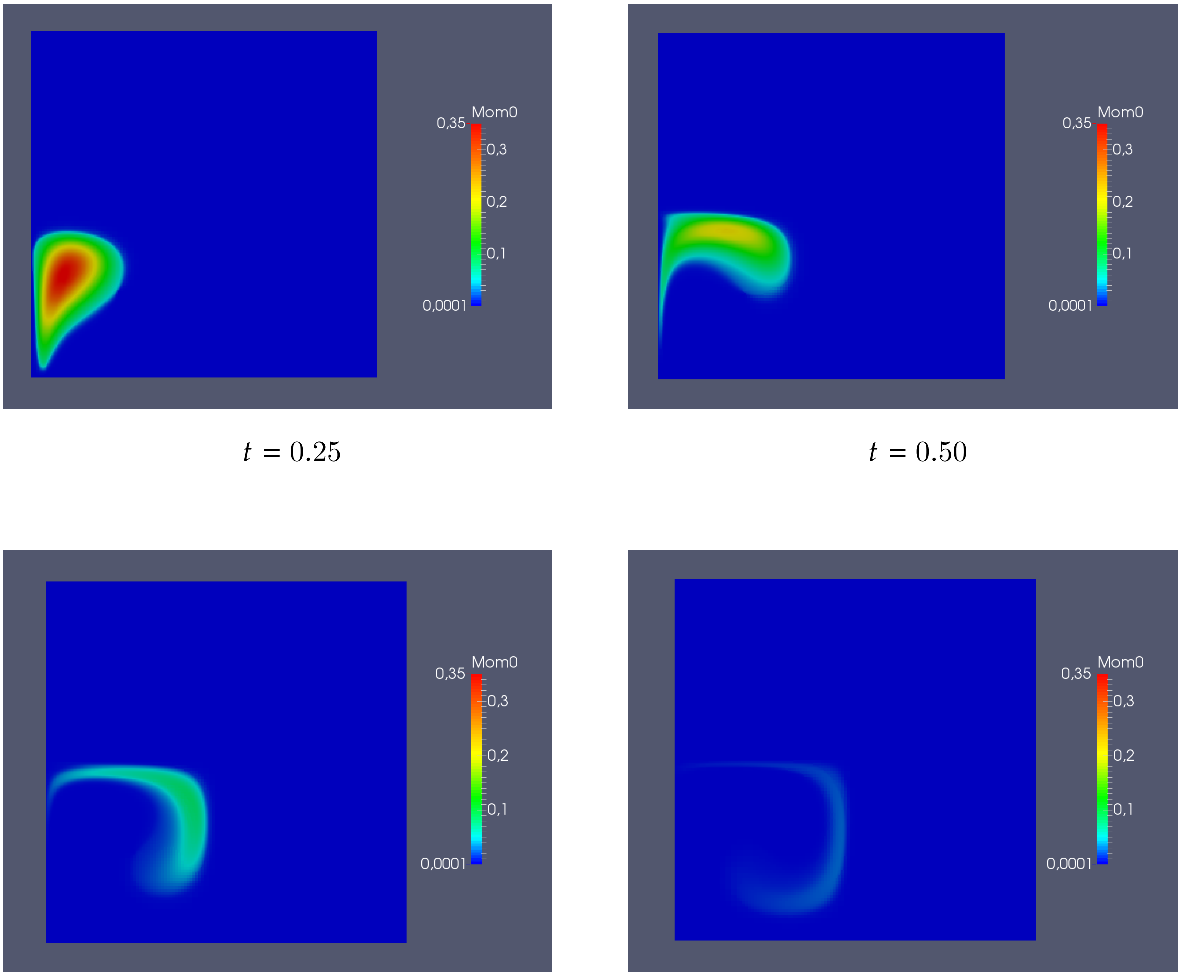


Figure 9.19: *Strong scaling of the Taylor-Green evaporated case in AMR grid, the maximum level is* $l_{max} = 9$ *and the minimum level is* $l_{min} = 4$

MPI process number	Mesh adaptation	Transport solver	Source term solver
1	0.1%	2.5%	97.3%
24	0.7%	2.2%	97.3%
60	0.9%	2.2%	95.7%
84	4.8%	2.7%	91.6%
96	6.16%	2.8%	90.5%

Table 9.4: *The time ratio (in percent) spent at each operation: Transport solver and Source term solver (Evaporation +drag) for different MPI process number.*

different MPI process number. In Figure 9.19, we present the strong scaling of the various step of the solving process (mesh adaptation, transport solver, source term solver and the total time). Up to 60 processes, we observe a good scalability. Once again, let us insist on the fact that for 60 processes and considering the level of compression reached, we end up with about 100 cells per core, which is few but still has a good level of efficiency. The transport part has about 80% of efficiency and the source term scales vary properly considering that load balancing could be also a problem (`p4est` also provides the ability to integrate computational complexity in the the load balance algorithm, which has not been used in the present simulations). Increasing the number of processes to 96, we have a remarkable decrease of the scalability, which reaches 56% for transport resolution, and that can be explained by the very small number of cells per process, which is going to kill the efficiency in terms of communications. However, the total efficiency is still about 67%, which shows both the efficiency of `CanoP` and `p4est` as well as the proper choice of the numerical strategy.



$t = 0.25$ $t = 0.50$

$t = 0.75$ $t = 1.0$

Figure 9.20: *Evaporating Taylor-Green simulation using second order scheme for transport in AMR grid with* $l_{max} = 9$, $l_{min} = 4$ *and* $\xi = 5.e-7$.

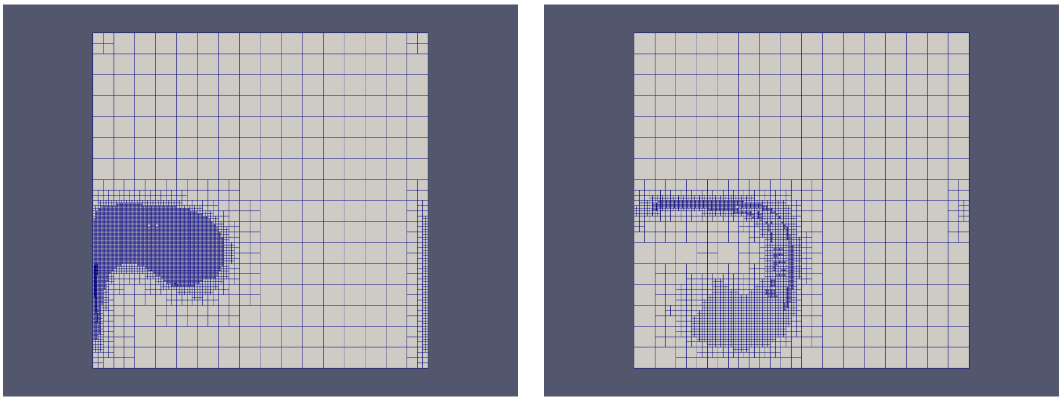


Figure 9.21: *The AMR grid in the case of Taylor-Green evaporating spray, with* $l_{max} = 9$, $l_{min} = 4$ *and threshold* $\xi = 5.e-7$.

9.6.3 Homogeneous isotropic turbulence in 2D/3D

In this section, we assess the code behavior in a more complex gaseous flow field representative of complex applications, where the spray is to be found everywhere in the domain. We consider a non evaporating spray respectively in a two- and three-dimensional configurations with a frozen Homogeneous Isotropic Turbulence (HIT) of the gas. The HIT gas field was generated independently by Sabat et al.(2014) with the ASPHODELE code of CORIA Reveillon and Demoulin(2007), that solves the three-dimensional Navier-Stokes equations for the gas phase under the low-Mach assumption.

2D HIT The characteristics of this HIT are given in Table 9.5, where Re_t is the turbulent Reynolds number, u_t is the velocity root-mean-square, ϵ is the mean dissipation rate, η_k is the scale of smallest structures, l_{int} is the integral scale of the turbulence, τ_k is the Kolmogorov time scale of the turbulence, and τ_{int} is the eddy turnover time. We consider a non evaporating spray. The initial spatial distribution is uniform.

Re_t	u_t	ϵ	η_k	l_{int}	τ_k	τ_{int}
7.12	0.1	0.01	0.022	0.1	0.36	1.0

Table 9.5: *Turbulence properties of the frozen HIT gaseous flow field.*

In order to limit the Particle Trajectory Crossings and to have comparable results between the monokinetic Eulerian simulation and the Lagrangian simulation, we consider low inertia droplets such that the Stokes number is taken $\mathrm{St} = 0.5$ based on τ_k. Numerical dissipation in the simulation of Eulerian models for spray dynamics is a key issue, since we have to use very robust numerical schemes due to the presence of singularities and asymptotic limits (zero density), but still want an accurate resolution in the smooth regions. In particular, numerical dissipation has a negative effect in predicting properly segregation of the spray in turbulent flows.

In order to limit this diffusion, we use the second order scheme and AMR in order to reach an accurate resolution in the regions of high concentration, while limiting the computational cost and memory trace. Figure 9.22 shows the spatial distribution of the number density at dimensionless time $t = 5$ (this dimensionless time corresponds to $t = 1.8s$ in the real time, because the characteristic time in this case is $\tau_k = 0.36s$). The droplets are ejected from vortices and concentrated in zones of low vorticity, thus, they segregate in regions of low vorticity. The segregation is computed using the spray moment m_0 as follows:

$$\frac{<(m_0)^2>}{<m_0>^2} \quad \text{where} \quad <\bullet>=\frac{1}{V}\int_V \bullet \quad dV.$$

In Figure. 9.23, we compare the time evolution of the segregation for a Lagrangian simulation (see Sabat et al.(2014)) and two Eulerian simulations :

- on uniform grid of 1024×1024 cells;
- using AMR with $l_{max} = 10$ and $l_{min} = 6$.

In both Eulerian simulations, we make a small error on the segregation compared to the Lagrangian simulation. However, the three results stay closely comparable. Furthermore, the segregation evolution of the two Eulerian simulations is closely the same. Therefore, we preserve

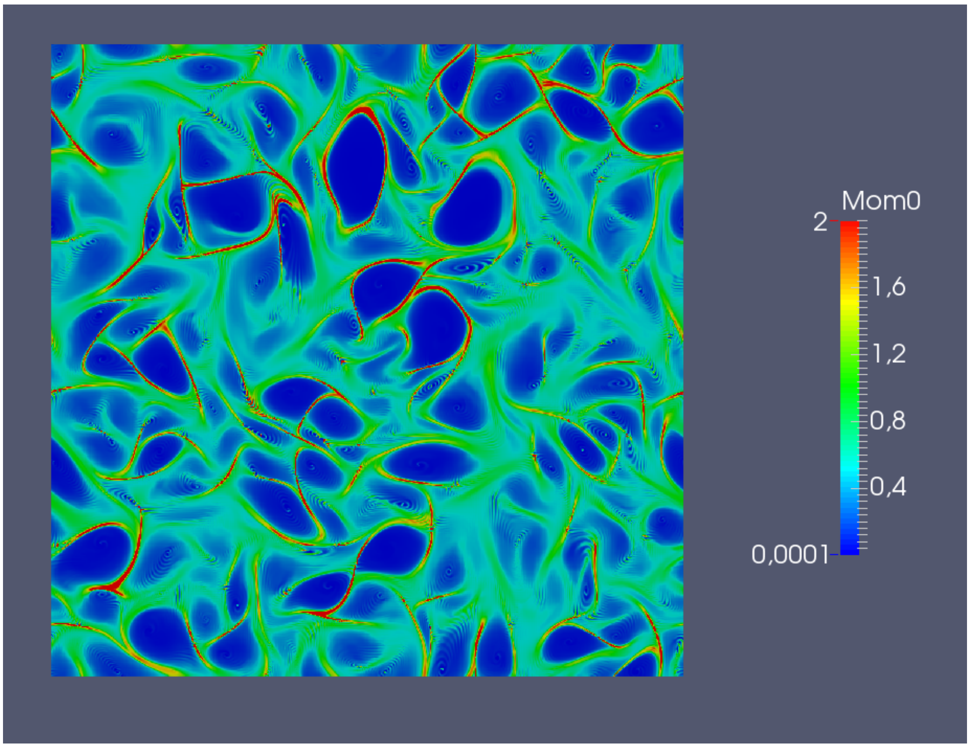


Figure 9.22: *Number density* m_0 *at* $t = 1.8$ *with AMR (* $l_{max} = 10$ *and* $l_{min} = 6$*) and with threshold* $\xi = 5.e-7$*.*

the same solution quality while using less cells in AMR simulation. We have reduced the computational time by a factor of two compared to the uniform grid computation and we obtain a compression rate of 55%. Let us underline that the chosen configuration is probably the worst configuration for such a strategy. However, we wanted to investigate the ability of the proposed strategy to cope with such a case, knowing that in realistic configurations, the flow is going to be turbulent, whereas the spray is going to occupy only a fraction of the computational domain.

3D HIT A 3D simulation of the spray in the presence of a frozen homogeneous isotropic turbulence was performed to test the AMR capacity in compressing the solution. The characteristic of the HIT is given in Table 9.6. The mean Stokes number of the polydisperse spray is taken as St = 0.5.

Re_t	ϵ	η_k	τ_k	TKE
25	$1e-3$	$3.2e-2$	1.	$1.e-2$

Table 9.6: *Turbulence properties of the frozen HIT gaseous flow field in 3D.*

The simulation is performed with an adaptive mesh, where the maximum and minimum refinement levels, $l_{max} = 9$ and $l_{min} = 7$, correspond respectively to a mesh of size $\delta x_{min} = 2^{-9}$ and $\delta x_{max} = 2^{-7}$ for a dimensionless grid $[0, 1]^3$ (the real physical domain dimension is $5m\times5m\times5m$). In Figure 9.24, we display a three cut sections of the spray density, as well as the corresponding mesh in Figure 9.25. The first figure shows the segregation of the spray in low vorticity regions. For the present choice of refinement criterion, the compression rate is 91.56%. In Figure 9.26,

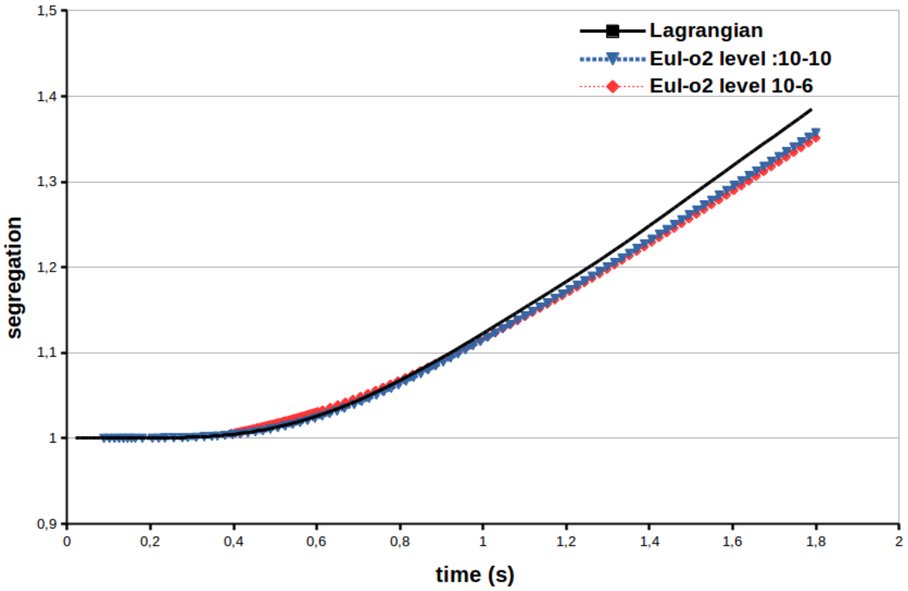


Figure 9.23: *Evolution of the segregation with time for the Lagrangian (black line), Eulerian on uniform grid* 1024×1024 *(blue line) and Eulerian with AMR (red line).*

we compare the segregation evolution obtained in this Eulerian simulation with a Lagrangian result (see Sabat(2016)).

We can see that in the AMR Eulerian simulation, we make a small error in the prediction of the segregation compared to the Lagrangian simulation because of the numerical diffusion. We underline that the monokinetic Eulerian model can also lead to overestimate the segregation of droplet, especially for high Stokes number. But here, the segregation computed from the Eulerian simulation is lower than the one of Lagrangian simulation, which is due to the numerical diffusion.

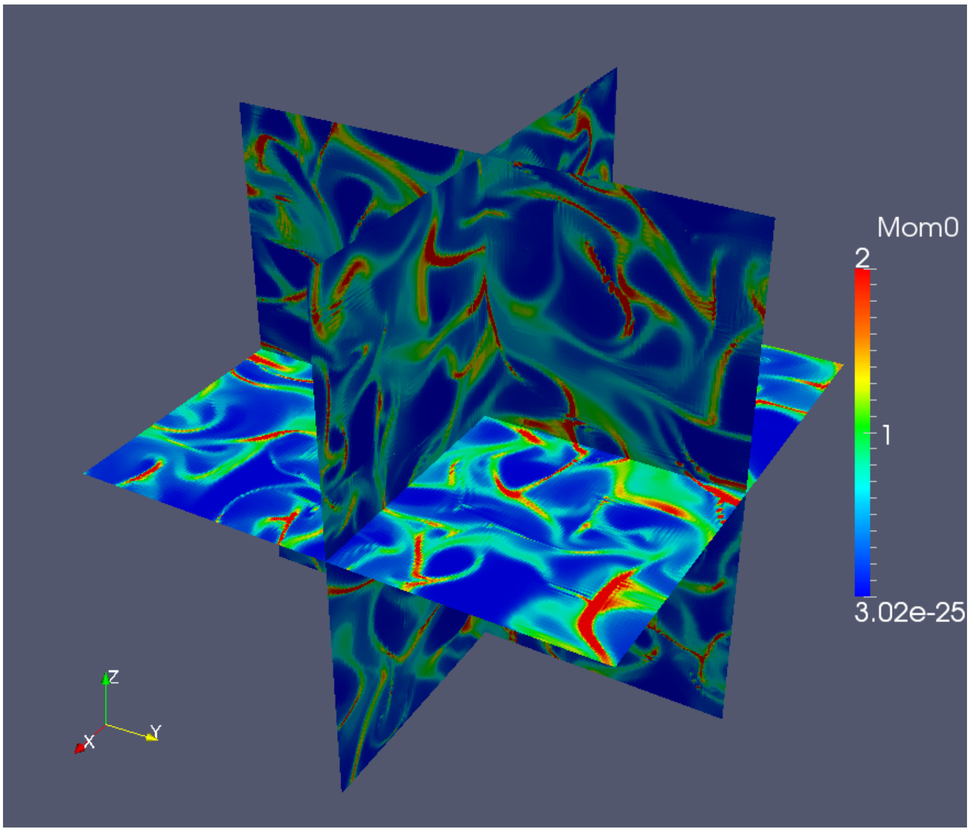


Figure 9.24: *Number density of the droplets given by the moment m_0, on an AMR grid at $t = 12$.*

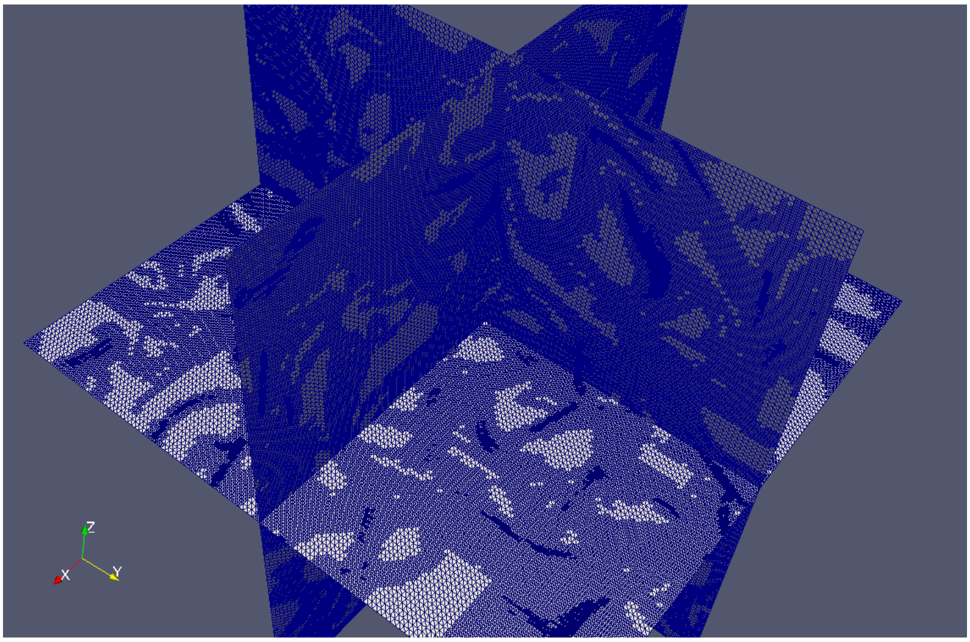


Figure 9.25: *The AMR grid at $t = 12$.*

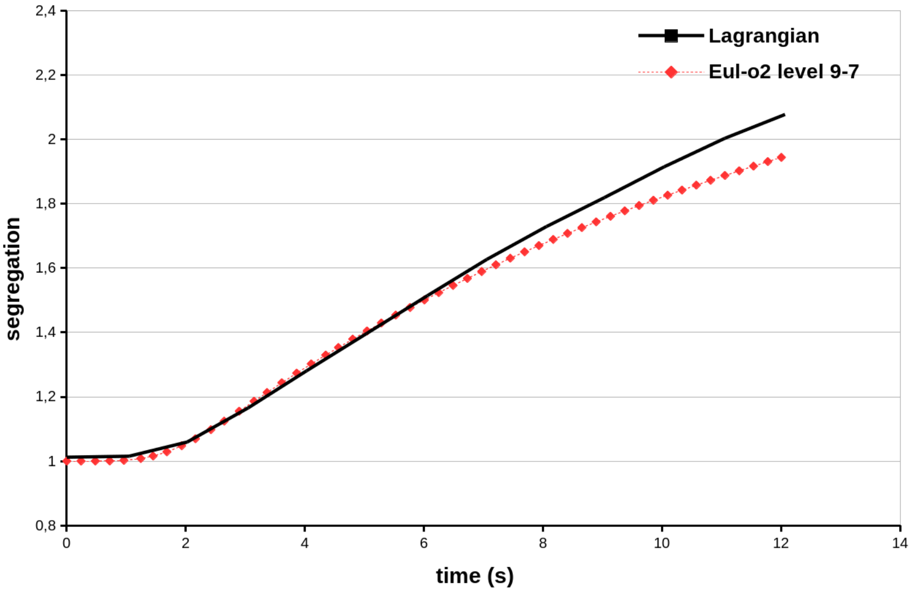


Figure 9.26: *Evolution of the segregation with time for the Lagrangian simulation (black solid line) and our Eulerian model with AMR (red dotted line).*

9.7 Conclusion

The aim of this chapter was first to present `CanoP` AMR code, which is based on the `p4est` library for mesh management in parallel computation. The main feature of this code is its flexibility and easiness to implement new finite volume applications and benefit from the efficiency of the `p4est` library with less programming effort. Different applications have been already implemented and tested. In this chapter, we also show how to extend the kinetic finite volume scheme in non-conforming meshes used here for the numerical resolution of the fractional moments model to simulate an evaporating polydisperse spray. A set of numerical tests, using the fractional moments model presented in chapter 4, have been performed in `CanoP` code to validate the numerical strategy and the parallel performance. The implementation of more complex and physical models with the possibility to couple a spray model with a two fluid model for the simulation of the whole process injection is envisioned in future works. In the present work, we show that we can reach a level of accuracy for AMR solution equivalent to uniform-mesh solution with lower computational time and memory. We also show a good scaling in parallel computation, especially for high workload per-cell compared to the communication and mesh management time. This is the case for the second order scheme for the transport and the source term that requires high arithmetic intensity. In future works, we aim to go further in developing high order schemes using Discontinuous Galerkin (DG) method. This method uses local degrees of freedom and do not require large stencil to achieve high order, which makes the method more attractive for AMR, since other methods that use large stencil may be expensive due to the need of recursive research of neighboring cells.

Chapter 10

The StarPU Runtime scheduler and the acceleration of the source term computations

This chapter is based on the work of Essadki et al.(2017), during CEMRACS 2015 summer school

10.1 Introduction

Nowadays, multi-cores machines are equipped with different Processing Unit (PU) types. We classify these devices into CPUs and accelerators (GPUs, Xeon-Phi, etc). Accelerators can accelerate numerical computations that involve intensive arithmetics. GPUs, for instance, are massively parallel devices, that can run Single Instruction Multiple Data (SIMD) programs very efficiently. Consequently, they can speed up the execution of tasks that perform intensive arithmetic operations on multiple data points simultaneously. However, it is not obvious to take advantages from these technologies without taking into consideration the architecture heterogeneity of the new generation of multi-cores machines. They involve different types of PUs and non uniform memory access. Indeed, approaching the theoretical performance of these architectures is a complex issue. The high performance computing community needs to face this challenge and propose reliable solutions that can allow other scientific communities to benefit from the computing power of these technologies. Many research efforts have focused on enhancing existing solutions: programming languages, compilers and libraries that can take into consideration accelerator-based machines. However, these solutions rely on a statistic distribution of the work among the processing units that is generally determined manually by the user. Another recent possible solution consists in using a runtime scheduler. Its role is to distribute the tasks of a program among the available computation resources (CPUs and accelerators) and to manage the memory, following a strategy that can optimize the CPU time and memory consumption.

StarPU Runtime Scheduler Augonnet et al.(2011) is a good candidate to provide these performances. In this chapter, we present the main results of the HODINS (High Order DIscontinuous methods with ruNtime Scheduler) project conducted during the CEMRACS 2016 summer school. During this project, we experienced **StarPU** in a 2D code, where we have solved numerically hyperbolic equations with possible source terms using Runge Kutta Discontinuous

Galerkin (RKDG) methods, which have been presented in chapter 7. The choice of Discontinuous Galerkin (DG) method is guided by its arbitrarily high order accuracy, its compactness and its memory locality. Two models have been separately implemented and tested in this 2D code: the Euler equations and the fractional moments model for polydisperse and evaporating sprays (see chapter 4 for more details). The main objective of this chapter is to illustrate a new solution that can accelerate the numerical simulation of complex models, similar to the one of the fractional moment model (chapter 4), thanks to a dynamic offloading of some tasks in GPUs, while using the numerical framework that has been presented in chapters 7-8. In this chapter, we briefly present the main features of **StarPU** and the implementation of the RKDG in a task-based program using **StarPU** library. Finally, we perform a set of numerical tests to first validate our implementation, assess and analyze the granularity of the tasks compared with the task overhead, the parallel performance of the code and the acceleration of the source terms resolution thanks to an execution on GPUs.

10.2 StarPU

StarPU is a task-based programming library for hybrid architectures. Therefore, the numerical method should be formulated in terms of tasks that can be executed in parallel. In StarPU, the tasks of an application are translated in a Direct Acyclic Graph (DAG) of tasks, where the nodes represent the tasks and the edges between the nodes express the dependencies between the associated tasks. The dependency is expressed implicitly through read/write access mode for each data memory i.e. for example if we submit successively two tasks "T_1" and "T_2", which perform some operations on the same data memory, such that the task "T_1" reads the data while the task "T_2" writes in the same data, the task "T_2" then depends on the task "T_1". After specifying the attributes of a task (data to handle, kernel functions, access mode to data, ects.), the function *startpu_insert*() is used to submit the task. The task insertion is asynchronous, where the tasks can be inserted each one after the other even if the first task is not yet executed. Following the order of task submissions and data dependencies, **StarPU** generates DAG of tasks and then distributes the tasks among the available computational resources

10.2.1 StarPU tasks

A **StarPU** task is made of:

- `kernels` are the functions that will be executed on a dedicated architecture: CPU, GPU, etc. A task may have the choice between different kernels implementation and it is the role of the **StarPU** scheduler to distribute the tasks on the available processing units following a certain criterion (in general minimizing the global execution time).
- `data handles` are the memory managers. Each data handle can be viewed as the encapsulation of a memory allocation, which allows to keep trace of the action (read or/and write) of the task kernel on the memory layout. In particular, this allows to build the task dependency graph.
- a `codelet`, the task descriptor. It contains numerous information about the task, including the number of data handles, the list of available kernels implemented to execute this task (for CPU, CUDA (for GPUs), OpenCL, ects.) and the memory access mode for each data handle: "Read" (R), "Write" (W) or "Read and Write" (RW).

We illustrate the creation, definition and submission of a task in **StarPU** in the following example:

A. We first create an empty task:

Algorithm 2 Create a task

```
starpu_task *task_A = starpu_task_create();
```

B. We provide this task with the data buffers, which are encapsulated in data_handle **StarPU** object:

Algorithm 3 Example of encapsulating a data_handle buffer

```
starpu_data_handle_t data;
starpu_matrix_data_register(&data,STARPU_MAIN_RAM, ptr_data ,...);/* a method to
encapsulate the ptr_data in data StartPU structure*/
task_A->handles[0] = data;
```

C. Then, we define the codelet that contains the kernel functions, the number of data buffers and the mode of the access to the data_handle buffer:

Algorithm 4 Example to define a codelet

```
starpu_codelet_init(&cl_A);
cl_A.nbuffers = 1; /*in this example we consider only one buffer */
cl_A.modes[0] = STARPU_W; /*state of data_handle: STARPU_W is only for write,
STARPU_R is only for read and STARPU_RW is for read and write*/
cl_A.cpu_funcs[0]=cpu_kernel_A;/*precise the CPU kernel function to execute the task in
a CPU*/
cl_A.cuda_funcs[0]=cuda_kernel_A;/*precise the CUDA kernel function to execute the task
in GPU*/
task_A->cl=&cl_A;
```

D. The kernel functions can be written for different devices. Below, we give prototypes for CPU and CUDA kernels:

Algorithm 5 Kernel function CPU and CUDA prototypes

```
void cpu_kernel_A(void *buffers[], void *cl_arg);
void extern "C++" void cuda_kernel_A(void *buffers[], void *cl_arg) ;
```

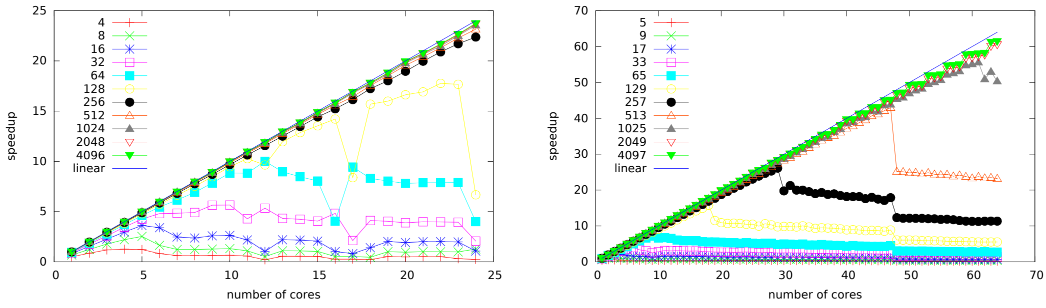
E. Finally we can submit the task as follows:

Algorithm 6 Submit a task

```
int submit_status = starpu_task_submit(task_A);
```

10.2.2 Schedulers

The purpose of the scheduler is to launch the tasks when they become ready to be executed. In StarPU, many different scheduling policies are available. In the present work, we consider only the two following:



2 dodeca-core Haswell Intel *Xeon Phi KNL*

Figure 10.1: *Task size overhead with the* `eager` *scheduler: scalability results obtained with duration of tasks varying between 4 and 4096 μs on two different machines. Reprinted from Essadki et al.(2017).*

- the `eager` scheduler: it is the default scheduler. It uses a central task queue. As soon as a worker has finished a task, the next task in the queue is submitted to this worker.
- the `dmda` scheduler: this scheduler takes into account the performance models and the data transfer time for the tasks. It schedules the tasks so as to minimize their completion time by carefully choosing the optimal execution architecture.

10.2.3 Task overhead

StarPU tasks management is not free in terms of CPU time. In fact, each task execution presents a latency called "*overhead*". We want to know when this additional time can be neglected or not. There is a **StarPU** benchmark that allows to measure the minimal duration of a task to ensure a good scalability. We send 1000 short tasks with the same (short) duration varying between 4 and 4096 μs and we study the scalability. In Figure 10.1, we plot the results obtained on a 2 dodeca-core Haswell Intel Xeon E5-2680 and on a Xeon Phi KNL with the scheduler `eager`. On few cores, we have a good scalability result, even if the duration of the tasks is short ($< 0.2ms$). On many cores, we need longer tasks duration to get satisfying scaling ($\approx 1ms$).

To sum up, if the duration of the tasks is smaller than the microsecond, their overhead cannot be neglected anymore.

10.3 RKDG task-based programming implementation

10.3.1 Principal task formulation of RKDG method

In this section, we present the essential steps in the implementation of a RKDG scheme to solve hyperbolic equations system with the source terms. The implementation is done in a 2D code, where we solve numerically the Euler equations or the fractional moment model (see chapter 4). In the following, we present a generic implementation of the two models, while we focus only on the fractional moments model for the numerical tests. The two systems of equations can be written in the following form:

$$\partial_t \boldsymbol{U} + \partial_x \boldsymbol{\mathcal{F}}_x(\boldsymbol{U}) + \partial_y \boldsymbol{\mathcal{F}}_y(\boldsymbol{U}) = \boldsymbol{S}(\boldsymbol{U}), \tag{10.1}$$

where $\boldsymbol{U} \in \mathrm{R}^{\mathrm{nVar}}$ is the vector of conservative variables, nVar being the number of these variables (for Euler equations nVar $= 4$ and for fractional moments model nVar $= 6$), $\boldsymbol{\mathcal{F}}(\boldsymbol{U}) = \big(\boldsymbol{\mathcal{F}}_x(\boldsymbol{U}), \boldsymbol{\mathcal{F}}_y(\boldsymbol{U})\big)$ is the flux tensor and $\boldsymbol{S}(\boldsymbol{U})$ is a source term.

The discontinuous Galerkin discretization of the transport part is given in (7.28) and (7.36). We recall here the integral version of this scheme:

$$\begin{aligned}\sum_{j=1}^{(k+1)^2} \Delta x^2 \boldsymbol{\mathcal{M}}_{i,j} \frac{d\boldsymbol{W}_j^m}{dt} = {} & \Delta x \int_{\xi\in[-1/2,1/2]^2} \boldsymbol{\mathcal{F}}(\boldsymbol{U}_h(t, \Xi_m^{-1}(\boldsymbol{\xi})) \cdot \nabla_{\xi}(\phi_i(\boldsymbol{\xi}))d\boldsymbol{\xi} \\ & -\Delta x \sum_{e\in D} \int_{\xi=-1/2}^{1/2} \boldsymbol{\mathcal{F}}(\boldsymbol{U}_h(t, \widetilde{\boldsymbol{x}}_{m,e}(\xi)) \cdot \boldsymbol{n}_{m,e}\phi_i(\Xi_m(\widetilde{\boldsymbol{x}}_{m,e}(\xi)))d\xi.\end{aligned} \tag{10.2}$$

See chapter 7 for the definition of the different terms in this equation.

As we can see from this scheme, some terms can be precomputed in the beginning of a simulation. These terms are:

- 1D and 2D Gauss-Legendre quadratures: we use these two quadratures to evaluate the integrals involved in the scheme (10.2).
- 2D Gauss-Legendre-Lobbato (GLL) quadrature nodes: these points are used in the limitation procedure of the Zhang & Shu method Zhang et al.(2012) (see section 7.3.3 for more details).
- The values of the gradient of the polynomial basis, at 2D Gauss-Legendre quadrature points of the domain $\left[-\frac{1}{2}, \frac{1}{2}\right]^2$. Indeed, the first term in the RHS of equation (10.2) is an integrated using 2D 2D Gauss-Legendre quadrature and it involves the gradients of the polynomial basis.
- The values of the polynomial basis, computed at the 1D Gauss-Legendre quadrature points on the four edges of the domain $\left[-\frac{1}{2}, \frac{1}{2}\right]^2$ are used in the second of the RHS of equation (10.2).

These terms are precomputed before submitting any task and stored in 1D and 2D arrays.

In the following, we consider uniform and Cartesian 2D grids. First, we compute the initial solution in each cell node. The number of degrees of freedom $(k+1)^2$ in a cell is given by the numerical order of the DG discretization $p = k+1$, where k is the polynomial degree used in the DG method. The temporal integration is preformed through an explicit Runge-Kutta method. Each step of Runge-Kutta method is updated as follows:

A. First, we check if the time step Δt satisfies the CFL condition (7.40). In the actual version of the code, the time step is fixed by the user. For future work, we will use a variable time step, which satisfies the CFL condition.

B. We compute the first part of the Right Hand Side (RHS) of the scheme (10.2), which corresponds to the integral in a cell of the flux $\boldsymbol{\mathcal{F}}(\boldsymbol{U}_h)$ multiplied by the gradient of the polynomial basis elements. This integral is approximated by a 2D Gauss-Legendre quadrature approximation.

C. For each cell, we interpolate the solution at the 1D Gauss-Legendre quadrature nodes of the edges of the cell. After performing this interpolation for all cells, we obtain two values for each node in a given edge: those are used to compute a numerical flux. For the fractional moments model, we use the upwind numerical flux described in (7.31) and (7.15).

D. For each cell, we compute the last part of the RHS of equation (10.2), by integrating the numerical flux multiplied by the polynomial basis elements over each edge of the computational cell. For this purpose, we use the numerical fluxes computed in the previous step.

E. The RHS of (10.2) is now completely computed. The ODE system (10.2) can be then integrated according to the corresponding Runge-Kutta step.

F. In the case of the fractional moments model with evaporation source term, we apply the modified limitation procedure of Zhang & Shu in the solution at 2D Gauss-Legendre quadrature points[1] for each cell as described in appendix C.

G. The source term is integrated at each degree of freedom see section 8.3.

H. We interpolate the solution at the GLL points. Then, we apply the limitation method of Zhang & Shu at these points (see section 7.3.3 for more details).

10.3.2 Data buffers and memory allocation

Since the tasks dependency graph is based on the memory dependency between successive tasks, memory allocation is a crucial part of our application. This starts with the definition of the following two structures: `Cell` and `Edge`. The `Cell` structure corresponds to internal variables in each computational cell and which contains an array of nDoF×nVar variables: nVar conservative variables per cell node and nDoF degrees of freedom. For the Euler model, the conservative variables are the mass density, the two momentum components and the total energy thus nVar = 4, while for the fractional moments nVar = 6 and the conservative variables are the four fractional moments and the two components of the velocity weighted by the surface area density of the droplets: $(\mathrm{m}_{0/2}, \mathrm{m}_{1/2}, \mathrm{m}_{2/2}, \mathrm{m}_{3/2}, \mathrm{m}_{2/2}u, \mathrm{m}_{2/2}v)$ (see chapter 4 for the definition of the fractional moments). The `Edge` structure contains `nEdge` × nVar conservative variables. This structure corresponds to the interpolated solution in each edge. Finally, we summarize the principal `Cell` and `Edge` variables used in our implementation as follows:

- `u`, the **computed solution** is a `Cell` object buffer. In each mesh cell, it contains nDoF × nVar floats, where $\mathrm{nDoF} = (k+1)^2$ is the number of degrees of freedom for each cell, $k = p - 1$ is the polynomial order in DG discretization and p is the theoretical order of accuracy.
- $\mathtt{u}_0$, the **copy of the computed solution** is a `Cell` object buffer. It copies and saves the contents of `u` at the beginning of each time iteration. The value $\mathtt{u}_0$ does not vary during the internal steps of the Runge-Kutta method.
- `RHS`, the vector of **residuals** is a `Cell` object buffer. This variable corresponds to the Right Hand Side (RHS) of equation (10.2) multiplied by Δt. At each Runge-Kutta stage step $1 \leq s \leq k+1$, the update is done by:

  $$\mathtt{u} = a_s^{(k+1)}\mathtt{u}_0 + b_s^{(k+1)}(\mathtt{u} + \mathtt{RHS}) \tag{10.3}$$

  where $a_s^{(k+1)}$ and $b_s^{(k+1)}$ are the Runge-Kutta coefficients given in the table 10.1.

---

[1] We call it here a modified limitation procedure, because normally the limitation of Zhang & Shu is located at the GLL points to compute the coefficient θ (see chapter 7), but here we use directly the Gauss-Legendre points which corresponds to the degrees of freedom. Thus, we obtain a realizable degrees of freedom which can be integrated to solve the source terms.

k	$a^{(k+1)}$	$b^{(k+1)}$
0	0	1.0
1	$(0., 0.5)^t$	$(1., 0.5)^t$
2	$(0., 3/4, 1/3)^t$	$(1, 1/4, 2/3)^t$

Table 10.1: *Strong Stability-Preserving Runge-Kutta coefficients.*

- `e`, the vector of **edge solution** is an `Edge` object array. It corresponds to the right and left values of the solution in 1D (k+1)-Gauss-Legendre quadrature nodes over each edge. The solution at these points is computed by interpolating `u` at the edges of the cell, as described in C. Each edge element contains `nEdge` × nVar floats, where `nEdge` $= 2(k+1)$.

The domain is decomposed into `NPartX` × `NPartY` sub-domains. Thus, allows to define tasks per sub-domain that can be executed on parallel. The `Cell` and `Edge` buffers are decomposed per subdomain. Typically, for each of the `NPartX` × `NPartY` subdomains (see Figure 10.4), we create local vectors for each of the four buffers defined above that are encapsulated in a **StarPU** data handle to follow the memory dependency.

Overlap additional memory buffer In order to minimize the communications and the dependencies between subdomains, each subdomain comes with four additional one-dimensional buffers corresponding to the four possible overlap-data needed at the subdomain boundaries (East, North, West and South), as illustrated in Figure 10.3.

Two vectors of size `NxLoc` × `sizeof(Cell)` (`ovlpS` and `ovlpN`) and two vectors of size `NyLoc` × `sizeof(Cell)` (`ovlpE` and `ovlpW`) are always allocated, whether the overlap needs to be used or not. The reason for that is that the number of data handlers passed to a **StarPU** kernel needs to be constant. Therefore, when copying the overlap-data for example, all the overlap data handlers are passed anyway but nothing is done if they are not needed, like in the case of a periodic subdomain in one direction.

Of course, here lies a small communication optimization, when sending a task on another device, since some useless data is transfered. However, we think that the overlap tasks should be of negligible size compared to the task acting on the entire subdomains and should be mainly executed on the host node.

10.3.3 Description of the tasks

In the section 10.3.1, we have presented the essential operations needed by a single update of a Realizable RKDG scheme. Based on this general decomposition, here the list of tasks implemented in our application. Between brackets is specified the memory data handlers accessed by the task, with their respective access rights given between parentheses:

- `initialCondition`[`uLoc`($\mathbf{W}$)]: fills each subdomain solution with the initial condition.
- `checkTimeStep`[`uLoc`($\mathbf{R}$)]: computes the largest characteristic speed within the subdomain. In order to avoid gathering these time constraints globally, we only check that the fixed time step Δt initially given by the user respects locally the stability constraint.
- `Fill_Overlaps`[`ovlpE`($\mathbf{W}$),`ovlpN`($\mathbf{W}$),`ovlpW`($\mathbf{W}$),`ovlpS`($\mathbf{W}$),`uLoc`($\mathbf{R}$)]: each subdomain fills the corresponding neighbors overlap data vectors. Figure 10.4 depicts the `copyOverlaps` task for two domains.

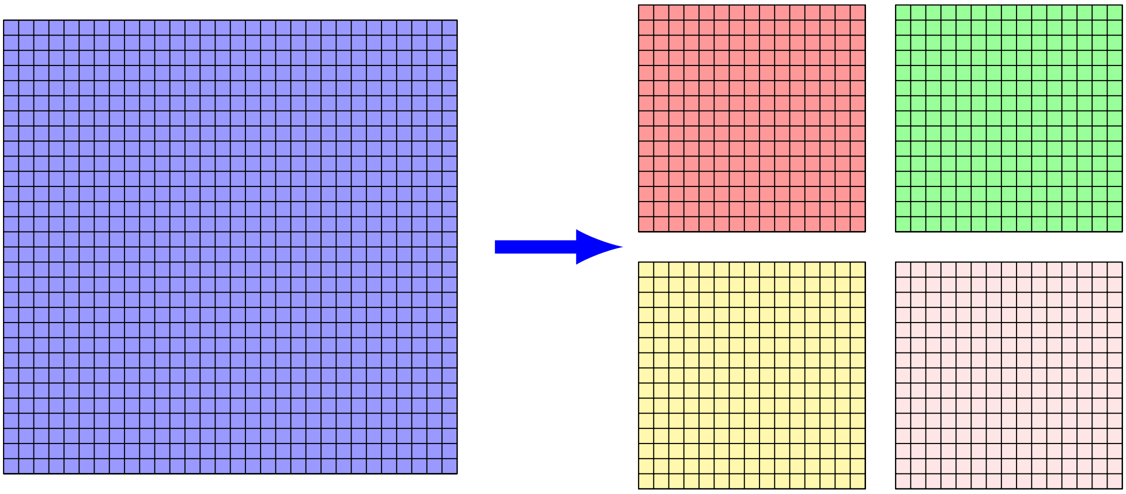


Figure 10.2: Partitioning of an initial mesh of $\{Nx = 30 \times Ny = 30\}$ cells into $\{\texttt{NPartX} = 2 \times \texttt{NPartY} = 2\}$ 225 cells domains.

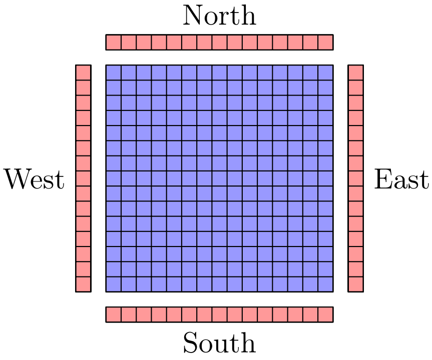


Figure 10.3: Partition (in blue) with its overlap (in red) in the East, North, West and South direction. Data of each partition is composed of these five handles.

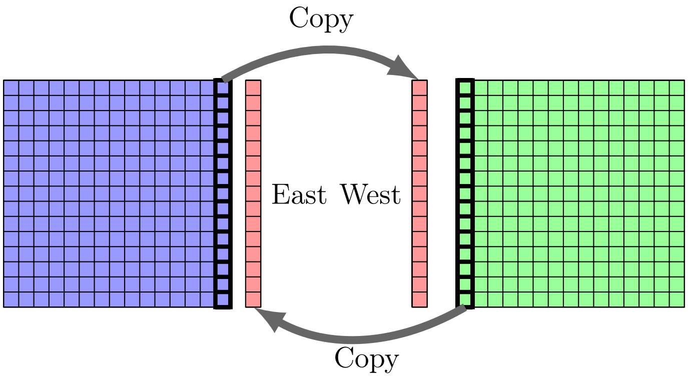


Figure 10.4: *Copy to overlaps tasks, example with two domains. The global computational domain is vertically divided into two parts, blue (left) and green (right). The green one is supplemented with a west overlap, whereas the blue one is supplemented with an east overlap. One residual computation includes two communications tasks: copying the right column of the blue domain into the west overlap of the green domain, and copying the left column of the green domain into the east overlap of the blue domain.*

- `interpolate`[`uLoc`(**R**), `eLoc`(**W**)]: for each cell, we interpolate the solution, given by the degrees of freedom and which is stored in `uLoc`, in 1D Gauss nodes over the four edges of the cell. We store the interpolated solution to the edges in `eLoc`.
- `InnerNumericalFluxes`[`eLoc`(**RW**)]: computes the numerical flux at 1D Gauss-Legendre quadrature points over the edges of the subdomain, except those of the boundaries. Finally, we store the numerical fluxes in `eLoc`.
- `project`[`uLoc`(**R**),`eLoc`(**R**),`RHS`(**W**)]: computes the residual at each 2D Gauss-Legendre quadrature points in the cells. The residual corresponds to the RHS of (10.2).
- `RK_update`[`uLoc`(**RW**),`RHS`(**R**)]: updates the numerical solution subdomain-wise, thanks to the update relation (10.3).
- `Source_term`[`uLoc`(**RW**)]: integrates the ODE system corresponding to the source terms.
- `Positivity`[`uLoc`(**RW**)]: applies the limitation procedure of Zhang & Shu to ensure realizable solution at GLL points.

The corresponding task diagram for one time step and two sub-domains is given in Figure 10.5. This graph is an output we can get from **StarPU** to verify the correct sequence of the tasks.

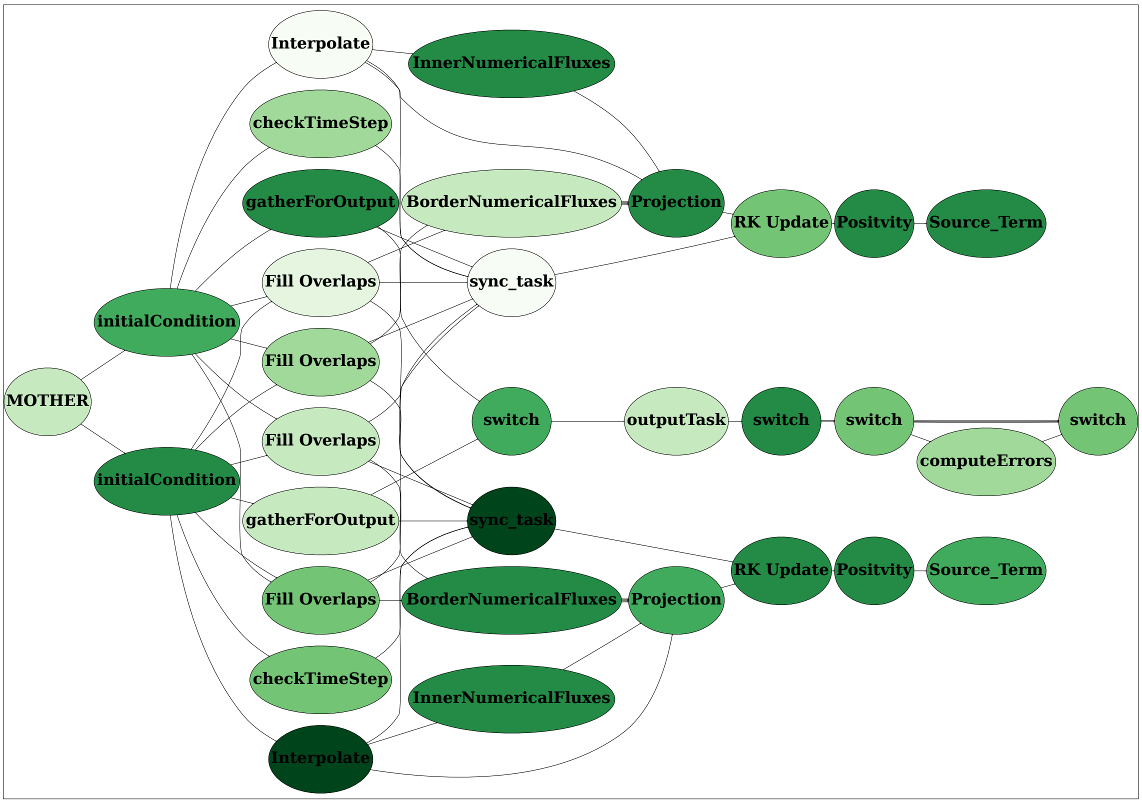


Figure 10.5: *Task diagram built by **StarPU** for one time iteration of our first order RKDG task-driven implementation on two subdomains, hence the horizontal symmetry.*

10.4 Results

In this section, we study the numerical accuracy of the RKDG scheme and the performance of our task-driven implementation on test cases solving the two-dimensional fractional moment equations over a 2D periodic domain. We perform two numerical test cases. The first one is a simple advection with a constant velocity and without source terms and the second one corresponds to an evaporation spray in the presence of a steady gas flow: Taylor-green vortices. In the first test case, we study the accuracy of the first and the second order RKDG scheme in a 2D domain. Using the advection test case, we perform a strong scaling study and we investigate the influence of the granularity of the tasks on the parallel efficiency. Finally, we use the second test case to illustrate the benefit of scheduling the tasks between CPUs and accelerators (GPUs). We show that the integration of source terms is very CPU time consuming, when its computation can be remarkably accelerated ton GPUs.

10.4.1 Mesh convergence study

We consider an advection of fractional moments with a constant velocity vector $(u, v) = (1, 1)$ in a periodic domain. The initial moments are expressed as follows:

$$m_{k/2}(t=0,\boldsymbol{x}) = \frac{2}{k+2}\left(S_{max}^{(k+2)/2} - S_{max}^{(k+2)/2}\right) r(\boldsymbol{x}-\boldsymbol{x}_c), \quad r(\boldsymbol{x}) = 1+\mathbb{1}_{(||\boldsymbol{x}||_2<0.25)}\cos^4(4\pi||\boldsymbol{x}||_2), \tag{10.4}$$

where $\boldsymbol{x}_c = (0.5, 0.5)$. The initial solution corresponds to the moments of rectangular size distribution $n(t, S) = \mathbb{1}_{[S_{min},S_{max}]}(S)$ multiplied by a spatial distribution given by $r(\boldsymbol{x} - \boldsymbol{x}_c)$. The

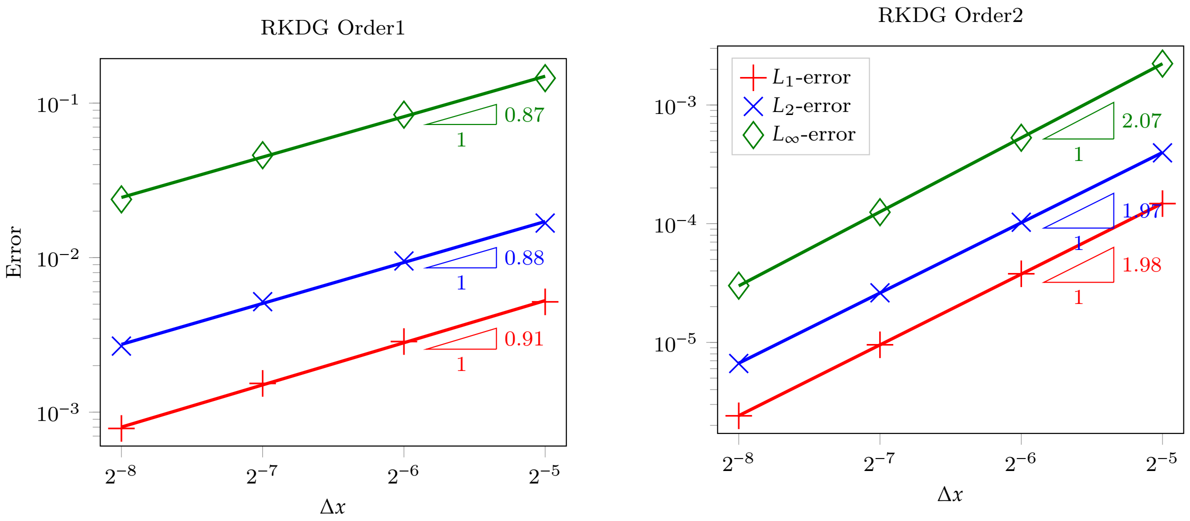


Figure 10.6: *Convergence curves for m_0 with respect to the grid refinement in logarithm scale. On the left RKDG of order 1 and on the right RKDG of order 2. Errors are computed using norm L^1 (plus), L^2 (cross) and L^∞ (diamond).*

exact solution is a simple advection at the initial uniform velocity $(u, v) = (1, 1)$. In Figure 10.6, we show the convergence results obtained with RKDG of order one and two. In this figure, we display in a logarithmic scale the L^1, L^2 and L^∞ numerical error depending on Δx. We obtain an order accuracy close to the theoretical order. For the RKDG of order one, the numerical order computed from the error solution of $m_{0/2}$ is around 0.9 for the three norms. For the second order moment, the order accuracy for $m_{0/2}$ is around 2.0 for the three norms. This validates the implementation of our numerical method up to the second order.

10.4.2 Parallel efficiency

In this section, we perform a strong scaling study using the same test case. We recall that the source terms are still not activated here. In the following, we keep the same problem size 256×256 cells and we increase the number of cores. We analyze the parallel efficiency of our implementation and its dependency on the domain partition number and the size of the tasks. Indeed, it is important to have a sufficient number of domain partitions, at least the number of cores, in order to allow **StarPU** to distribute the tasks among the available CPU resources and perform some scheduling optimization that can for example reduce the computational time depending on the scheduler policy. The second point concerns the size of the tasks. As discussed in the section 10.2.3, the task management by **StarPU** presents a time overhead. To obtain a good scalability, the task duration should be much larger enough than the overhead time. To sum up, in order to scale correctly on many cores, we need a sufficient number of long duration tasks. In Figure 10.7, we display the speedup curves of parallel computation for different numbers of domain partition `NPart`: using the first and the second order RKDG schemes. In this figure, we distinguish three situations. First, the cases for which we do not provide enough tasks `NPart` < 24. In these cases, the parallel computation does not scale correctly (`NPart` $= 1$and9). The second type of curves corresponds to large number of partition: `NPart` $= 256, 576$ (for first order RKDG scheme) and `NPart` $= 576$ (for second order), which are saturated due to task overhead. And finally, the cases where we use the right number of partition that allows to obtain a good scaling: `NPart` $= 25, 64$ (resp `NPart` $= 25, 64, 256$) for the first order (resp second

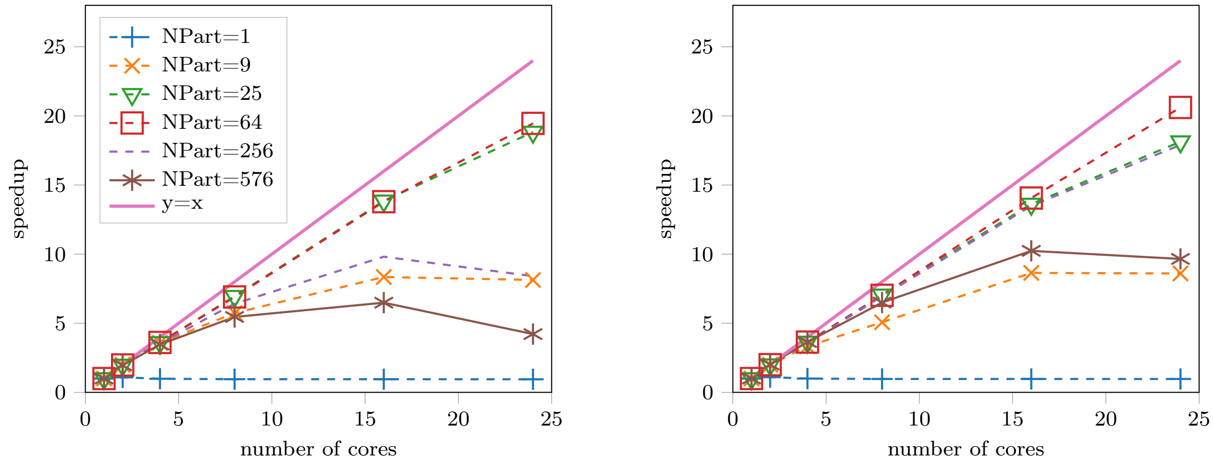


Figure 10.7: *Strong scaling with eager scheduler: scalability results obtained on 2 dodeca-core Haswell Intel Xeaon E5-2568 architecture using first order scheme (left) and second order scheme (right) with* `NPart` *varying from* 1 *to* 576.

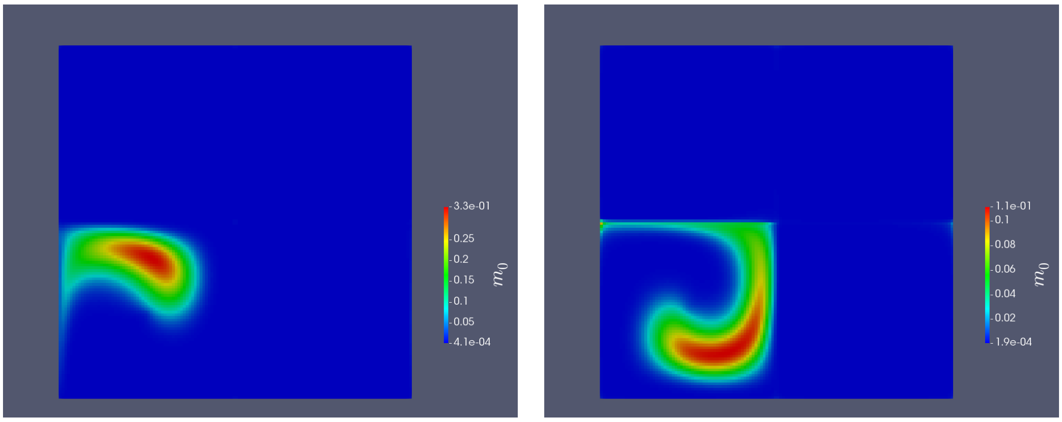


Figure 10.8: *The numerical solution of the evaporating droplets number density using the second order RKDG scheme: at* $t = 0.5$ *(left) and* $t = 1.0$ *(right).*

order). The granularity of the tasks does not only depend on the number of cells but also on the amount of the work in each cell. That explains why the first order saturates faster than the second order when we increase the number of partitions.

10.4.3 Source terms acceleration through GPU

In this section, we use the same numerical test case as the one described in section 8.4.3. The numerical resolution is achieved through a second order RKDG scheme with the algorithm described in 8.3 to integrate the source terms (drag force and evaporation). Zhang & Shu method is activated to ensure realizable a solution at Gauss-Legendre-Lobbato and Gauss Legendre points. Figure 10.8 shows the numerical results at two different times.

First, we evaluate the execution time of the different tasks in CPUs. Figure 10.9 shows a Gantt chart of the time duration of each task for 5 iterations and the distribution of the tasks between four CPU cores.

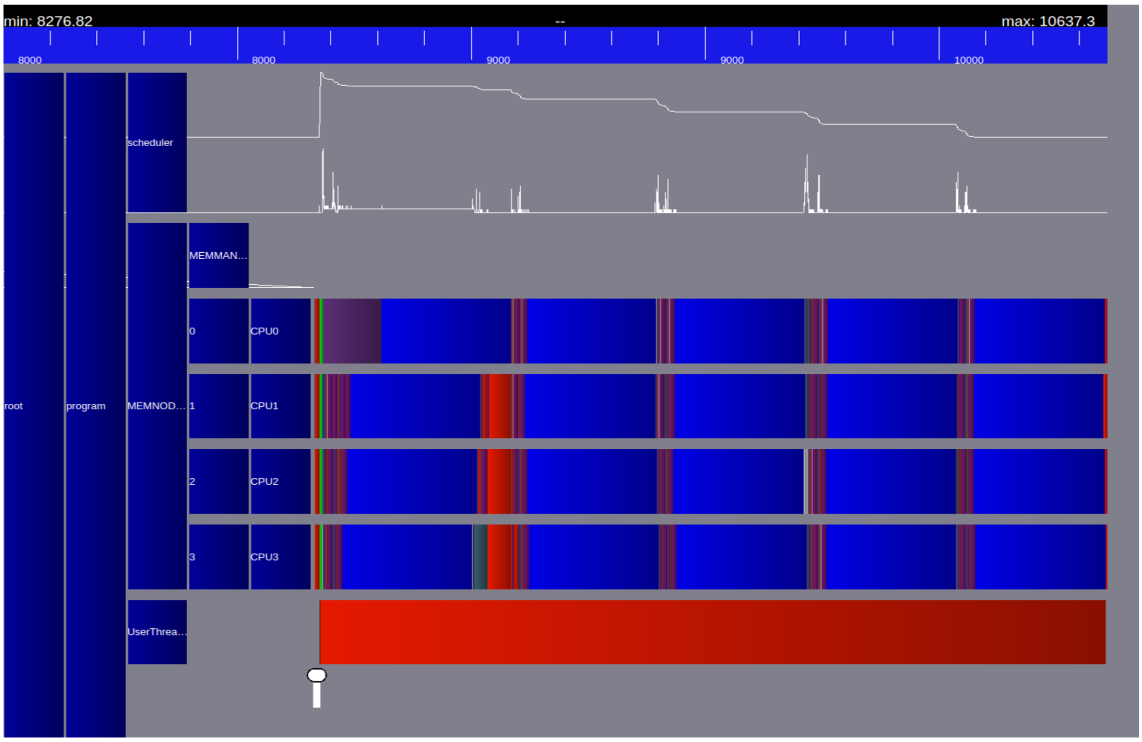


Figure 10.9: *Gantt chart for 5 time iterations of the second order RKDG scheme on four CPUs and using* `eager` *scheduler. Source term tasks correspond to the blue stripes, the red stripes correspond to slipping time and the other colors correspond to the other tasks.*

It is clear that the execution time of the source term tasks are mush larger than the other task ones. The source term tasks require intensive arithmetic operations because of the reconstruction of the size distribution by entropy maximization. Accelerator devices can accelerate the execution and reduce the computational time in these cases. In the following, we test the execution time of 50 iterations on a 256×256 mesh with or without GPU. Table 10.2 summarizes the computational time for different subdomain decomposition and using the eager or the dmda scheduler.

As it is expected, we see that activating the GPU immediately decreases the computational time for both `eager` and `dmda` schedulers. However, the `eager` scheduler is not smart enough to anticipate the global computational time of the tasks. Consequently, it distributes the tasks between GPUs and CPUs without taking into account the performance of each processing unit in executing the tasks. Using the `dmda` scheduler, we remark a gain in performance by better distributing the tasks on their best devices: this is illustrated on the Gantt charts in Figure 10.10 for both `eager` and `dmda` schedulers. On the top figure, we look at the task scheduling in CPUs and GPUs using the `eager` scheduler. First, we can see, as mentioned before, that the execution of source terms on the CPUs takes an important time while their execution in GPUs is much faster. The total execution time is reduced thanks to the GPUs but it could be reduced further with a smarter scheduler. For the `dmda` scheduler, the first tasks are distributed randomly in CPUs and GPUs since we do not initially provide a performance model of the tasks. But, after few iterations the `dmda` scheduler uses the performance of CPUs and GPUs on executing the first iterations to build a performance model of tasks execution on the different available PUs. For the last iterations, we see that the source terms tasks are mostly executed on the GPUs. This explains why, when turning the `dmda` scheduler on, the computational time is again reduced by

scheduler	part.	0 GPU	1 GPU
`eager`	2×2	529.608	549.166
	2×4	536.686	299.048
	4×4	530.345	193.645
	4×8	536.425	185.621
`dmda`	2×2	790.951	130.22
	2×4	679.207	92.6217
	4×4	564.299	100.019
	4×8	540.095	137.222

Table 10.2: *Computational time (in seconds) for* 50 *iterations on a* 256×256 *domain divided into* 2×2*,* 2×4*,* 4×4 *or* 4×8 *subdomains, when activating or not a GPU accelerating unit and using the* `eager` *or* `dmda` *scheduler.*

more than a factor of 2. Nonetheless, we also see that some red stripes corresponding to slipping time, still remain on the CPU line, meaning that one could gain even a little more by activating the computation of the transport tasks on the GPU card.

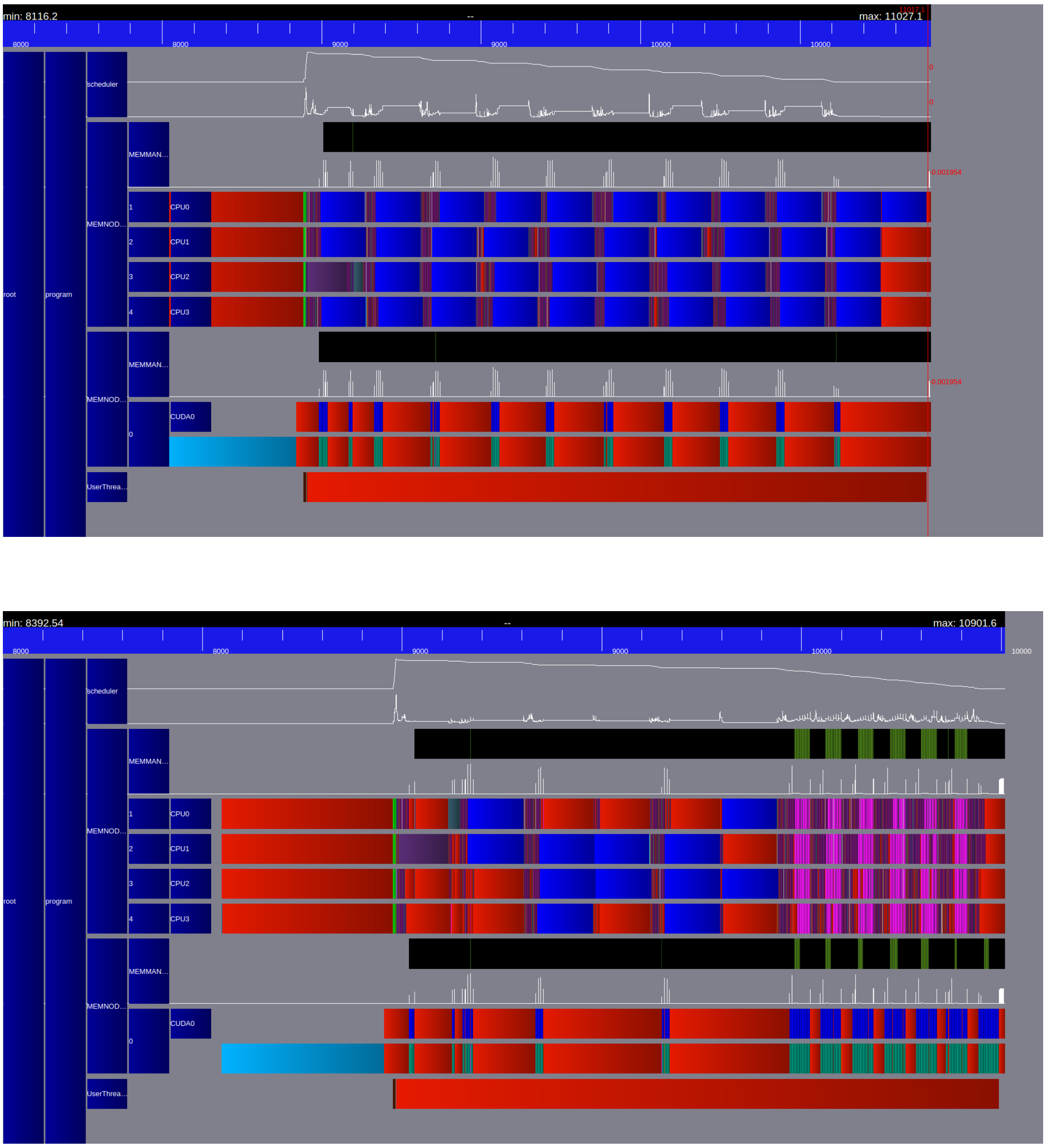


Figure 10.10: *Gantt chart for 10 time iterations of the second order RKDG scheme on four CPUs and one CPU. Source term tasks correspond to the blue stripes and the red stripes correspond to slipping time: using* `eager` *scheduler (top) and* `dmda` *scheduler (down).*

10.5 conclusion

In this chapter, we explore the possibility to reduce the computational time by taking advantages of nowadays heterogeneous supercomputer architectures. The proposed solution consists in implementing our numerical method in a task-based programming environment and in using a runtime scheduler to distribute the tasks among the different processing units (CPUs and accelerators) in a smart way. For this purpose, we use the **StarPU** runtime scheduler.

We have described and shown experimentally the main features of our code and the decomposition of the numerical method steps into different tasks. For the numerical resolution, we use on RKDG scheme with an arbitrarily high order implementation. This method is presented in details in chapter 7, where we show how to ensure a realizable and a robust solution, while the implementation in a task-based manner is described in this chapter. For this implementation, we validate the numerical method up to the second order. Then, we investigate the parallel scaling of the computations and its dependency on the number of domain decompositions and the size of the tasks. The main conclusion of this part shows that a correct scaling on many cores requires a sufficient number of tasks of long duration. Finally, we use GPUs to accelerate tasks that require intensive arithmetic computations. That is the case for source terms of the fractional moments model, but it can be generalized for other tasks that require an important workload. We also illustrate briefly the impact of the scheduler policy in distributing the tasks between the different PUs and in reducing the computational time. These encouraging results show a promising potential of tasked-based programming with a runtime scheduler to take benefit from the new hybrid-multicores machines and reduce the computational time of complex simulations.

General conclusion and perspectives

The studies conducted in this PhD thesis are dedicated to the numerical simulations of the fuel injection in combustion chambers and in particular automotive engines. In this context, we aim at developing new solutions that can enable the design of new combustion chambers through new models as well as robust and accurate numerical methods, which can simulate the whole injection process. At the same time, we need to satisfy some industrial requirements in terms of predictive simulation, but still ensuring an acceptable computational cost and a high performance computing on parallel architectures. The numerical modeling of the fuel injection flow raises a number of challenges and difficulties. Indeed, we have to deal with a two-phase flow that involves multi-scale processes. In this complex flow, many physical phenomena play an important role: the turbulence, the deformation of the gas-liquid interface, the breakup mechanisms, the polydisperse cloud of droplets and the interaction between the two phases (evaporation, drag force, etc.). Direct Numerical Simulation aims at solving all the involved scales and provides fine details about the flow. Therefore, it is computationally very expensive for industrial applications. For this reason, developing Eulerian reduced order models is a promising option for industrial applications. In this type of models, the equations are not solved for all the scales, but instead the small scale effects are taken into account through an appropriate sub-scale description, where a threshold scale has been chosen.

Up to now, the existing reduced order two-phase flow models can not simulate the whole injection process, instead each of these models depends on the flow regime: separated or disperse phases. In this PhD, we have considered two reduced-order models families. First, the two-fluid and homogeneous models are suitable to simulate the injection flow in the separated phases region. However, we have underlined two major difficulties: the artificial diffusion of the interface and the the lack of information at the interface sub-scales. Second, we choose the Eulerian moment approach to simulate the disperse phase and in particular the high order size-moment with a continuous reconstruction of the size distribution as it was done in the EMSM model Kah(2010). This Eulerian model shows good results to capture the polydispersion using only one size-section. However, it is important to point out that the EMSM model is restricted to spherical droplets and it is not yet adapted for a coupling with a separated phases sub-scale model, when the interface has complex topology. Therefore, to design a unified model based on these two models families (two-fluid and high order size-moment models) as well as robust and accurate numerical methods dedicated to highly parallel computation, we have addressed these issues in three main axes:

- **In terms of modeling**: we have proposed a new approach to design a unified model for both separated and disperse phases. Our strategy consists in introducing variables that can be defined for all regimes and can be used to enhance the sub-scale description of the interface and at the same time to describe the polydispersion in the disperse phase. Such strategy would allow to avoid a coarse coupling between two different models using different variables. In this perspective, we have introduced new concepts and tools to model the sub-scales:

  - First, we have defined a new distribution on surface, called SDF, to extend the distribution in number given by the NDF for the disperse phase. The phase space of this new distribution is decomposed of the mean and Gauss curvatures and the interface velocity. We also derived the link between the two distributions SDF and NDF. The SDF is defined independently of the flow regime and it can be related to the NDF in the case of a dilute disperse phase. Evolution equation of the SDF describes the evolution of the interfacial variables and the surface stretching. From this equation, we have derived a system of the SDF moments. We have also showed that the first order moments of the SDF are the averaged geometrical variables of the gas-liquid interface proposed in Drew(1990).
  - In the case of spherical droplets, we showed that the averaged geometrical variables are expressed as fractional size-moments of the NDF, where the size is given by the surface area of the droplets. We also showed that the different properties of integer moments already used for the EMSM model can be extended and used for the fractional moments model. In chapter 8, we showed that the new fractional moments model has the same capacity to describe the polydispersion as the EMSM model.
  - In chapter 6, we have classified the internal variables of the SDF into resolved and unresolved phase space. The moments of the SDF integrated only over the unresolved part of the phase space define the sub-scale geometrical moments. The equations satisfied by these sub-scale variables involve exchange source terms with the resolved scales. These source terms will have to be closed and related to the variables of a two-fluid model.
  - Finally, we have designed an algorithm to compute the curvatures from a level-set field dedicated to the post-processing of DNS two-phase flow simulations. This new algorithm preserves some geometrical and topological information, which essentially allows us to compute a NDF from an averaged SDF. The main objective of such post-processing is to provide a deep insight on the evolution of the flow and propose closure relations of the source terms involved in the system of equations of the sub-scale geometrical moments.

- **In terms of numerical schemes**: we have developed accurate and robust numerical schemes for the resolution of the fractional size-moment model dedicated to the numerical simulation of polydisperse and evaporating spherical droplets. In this context, we have adopted a discretization strategy based on an operator splitting method, where we separately solve the transport part, then the source terms of the system of equations.

  - For the numerical resolution of the transport part of the equations, we proposed two types of numerical method. The first one is based on a Finite Volume Kinetic (FVK) discretization de Chaisemartin(2009); Kah et al.(2012). From this approach, realizable and robust schemes up to the second order have been derived. The second one is the Runge Kutta Discontinuous Galerkin (RKDG) scheme Cockburn and Shu(1998). The realizability of this method is ensured through a limitation method developed in Zhang et al.(2012). The RKDG method is an arbitrary high order scheme. In our work, we limited the study to the third order. A comparison between the FVK and RKDG methods was conducted in chapter 7. The comparison shows accurate results of RKDG compared to FVK. However, FVK schemes are more robust and stable in critical situations such as delta-shock formation that appears in the case of the jet-crossing.

  - In chapter 8, we proposed an original ODE solver to solve the evolution of the moments due to the source terms. A new evaporation solver dedicated to the fractional moments is developed based on ideas similar to the ones proposed the first time in Massot et al.(2010). In fact, we use a continuous reconstruction through entropy maximization to evaluate the disappearance fluxes due to the evaporation and we combine a kinetic-based scheme with a Direct Quadrature Method of Moment to evaluate the evolution of the size. However, the new algorithm involves negative order moments. Thus, special treatments have been proposed in chapter 8 to deal with this difficulty.

- **In terms of scientific and high performance computing**: we have studied some solutions in order to save the computational CPU time and memory or to accelerate some intensive arithmetic tasks, while ensuring an accurate resolution and parallel computations in standard and heterogeneous architectures.
  - First, we implemented the fractional moments model in the `CanoP` code, which is based on the `p4est` AMR cell-based library. The choice of using cell-based AMR was initially determined by the high resolution needed for localized sharp interfaces, as in separated-phase flows Drui(2017). The results of a series of 2D and 3D simulations in chapter 9 showed that the use of AMR grids can preserve the accuracy of the solution and reduce significantly the computational time compared to the uniform grids. Furthermore, we showed a good parallel performance in CPUs machine thanks to the `p4est` library which is not obvious to obtain for an AMR grid.
  - On the other hand, we implemented the fractional moments model in a 2D task-based program using the StarPU library to schedule the tasks in heterogeneous architectures (CPUs and accelerators). StarPU scheduler distributes the different tasks among the available computation resources and manages the memory following a strategy that can optimize the consumption of CPU time and memory. The results showed an important acceleration of tasks that require intensive arithmetic computations and in particular the ODE evaporation solver. These encouraging results show a promising potential of tasked-based programs with a runtime scheduler to take benefit from the new hybrid-multicore machines and reduce the computational time of complex simulations.

The present PhD proposes new solutions for the numerical simulation of the fuel injection in terms of modeling, numerical method and high performance computing. The results obtained in these three major fields are encouraging to continue the investigation in these directions. Future works should provide appropriate closures for the sub-scale model proposed in chapter 6, ensure a robust and accurate resolution of this model and satisfy a good scalability in massively parallel computation using AMR grids. The future works can follow the following directions:

- First, in terms of modeling, closure expressions of the time evolution of interfacial variables (curvatures and interface velocity) should be derived by post-processing and analyzing realistic jet DNS results in different regions of the jet as it was illustrated in chapter 6. The statistical evolution of the curvatures and the interface velocity need to be closed and expressed as a function of the transported variables of the two-fluid model and the other sub-scales variables. Furthermore, we need to propose an appropriate reconstruction of the SDF or the multi-variable NDF. Closing these terms will allow to close the source terms involved in the system of equations of the sub-scale geometrical moments (see chapter 6 for more details). On the other hand, we will need to take into account the effect of the sub-scales on two-fluid models. A first contribution to enrich a two-fluid model with some micro-scale effects is proposed in the PhD of Drui(2017). The proposed model has been derived from the Hamilton's variational and the second thermodynamics principles Gavrilyuk and Saurel(2002), where a pulsation energy is used in the Hamiltonian formulation. The obtained two-fluid model takes into account some micro-inertia and micro-viscosity effects that are related to the pulsations of the interface. However, this model is restricted to spherical and monodisperse bubbles/droplets. In Di Battista(2020); Cordesse(2019) PhDs, a similar work aims at expressing some surface energies as a function of the sub-scale geometrical moments. Then, by using these energies in the Hamiltonian formulation, the authors aim at deriving a new two-fluid model that takes into consideration the sub-scale description given by the sub-scale geometrical moments.

- Second, in terms of the numerical implementation and the parallel computing, we will pursue the developments in `CanoP` code. The next step would be the implementation of the Galerkin Discontinuous scheme in AMR grid. The implementation in uniform 1D and 2D grids has been already investigated and showed very promising results. Now, we should adapt the scheme to non-conforming meshes. We will first implement this method in the fractional moments model for the disperse phase and the mixture two-fluid model Drui et al.(2016b) for separated phases. Furthermore, the improvement of the numerical accuracy also requires to control the compression error generated by the computations on adapted grids. More appropriate refinement strategies could be considered by using a multiresolution approach Harten(1994); Duarte(2011) or by using criteria based on the local truncation error as in Berger and Oliger(1984). Finally, developments of methods to enable the computation over heterogeneous architectures should be considered to benefit from the performance of the accelerators to run some specific tasks and also to ensure the parallel scaling in the new multi-core heterogeneous machines.

Appendix A

Realistic droplet models

The closure models used in Section 3.3 of the main paper are based on simplified assumptions. In this appendix, we demonstrate that the present contribution can be generalized to more realistic physical models. In this part, we consider a dilute spray of spherical droplets, where the droplets experience evaporation, drag and thermal transfer, and we neglect the collision, coalescence and fragmentation. In this context, WBE can be written as follows:

$$\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{c} f) + \partial_{\boldsymbol{c}} \cdot (\boldsymbol{F} f) + \partial_S \left(\mathrm{R}_S f\right) + \partial_T \left(\boldsymbol{e}^* f\right) = 0, \tag{A.1}$$

where the drag force $\boldsymbol{F}(t, \boldsymbol{x}, \boldsymbol{c}, S)$ depends on time, space, velocity and size. The evaporation rate $\mathrm{R}_S(t, \boldsymbol{x}, S, \boldsymbol{c}, T)$ and thermal transfer $\boldsymbol{e}^*(t, \boldsymbol{x}, S, \boldsymbol{c}, T)$ depend on the time, space, velocity, size and temperature.

We consider the following presumed NDF form:

$$f(t, \boldsymbol{x}, \boldsymbol{c}, T, S) = n(t, \boldsymbol{x}, S)\delta(\boldsymbol{c} - \boldsymbol{u}(t, \boldsymbol{x}, S))\delta(T - T_d(t, \boldsymbol{x}, S)) \tag{A.2}$$

The semi-kinetic equation can then be obtained by integrating (A.1) with respect to $(\boldsymbol{c}, T)$ after multiplying it by $(1, \boldsymbol{c}, e(T))^t$, where $e(T) = e_0 + \int_{T_0}^{T} C_{v,l}(T')\mathrm{d}T'$ and $C_{v,l}(T)$ is the liquid fuel heat capacity at constant volume:

$$\begin{array}{rcl} \partial_t n + \partial_{\boldsymbol{x}} \cdot (n\boldsymbol{u}) & = & \partial_S(\mathrm{R}_d n), \\ \partial_t n\boldsymbol{u} + \partial_{\boldsymbol{x}} \cdot (n\boldsymbol{u} \otimes \boldsymbol{u}) & = & \partial_S \left(n\mathrm{R}_d \boldsymbol{u}\right) + n\boldsymbol{F}_d \\ \partial_t n e_d + \partial_{\boldsymbol{x}} \cdot (n\boldsymbol{u} e_d) & = & \partial_S \left(n\mathrm{R}_d \boldsymbol{u}\right) + nC_{v,l}(T_d)\boldsymbol{e}_d^*, \end{array} \tag{A.3}$$

where $\mathrm{R}_d = \mathrm{R}_S(t, \boldsymbol{x}, \boldsymbol{u}, S, T_d)$, $\boldsymbol{e}_d^* = \boldsymbol{e}^*(t, \boldsymbol{x}, \boldsymbol{u}, S, T_d)$, $\boldsymbol{F}_d = \boldsymbol{F}(t, \boldsymbol{x}, \boldsymbol{u}_d, S)$ and $e_d = e(T_d)$.

In the following, we consider that the temperature and the velocity are independent of the droplet size. Then, we derive the moment governing equations from the semi-kinetic system:

$$\left\{ \begin{array}{ccccc} \partial_t \mathrm{m}_0 & + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_0 \boldsymbol{u}) & = & -n\mathrm{R}_d|_{S=0}, \\ \partial_t \mathrm{m}_{1/2} & + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{1/2} \boldsymbol{u}) & = & \mathrm{G}_{\mathrm{m}_{1/2}}, \\ \partial_t \mathrm{m}_1 & + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_1 \boldsymbol{u}) & = & \mathrm{G}_{\mathrm{m}_1}, \\ \partial_t \mathrm{m}_{3/2} & + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_{3/2} \boldsymbol{u}) & = & \mathrm{G}_{\mathrm{m}_{3/2}}, \\ \partial_t \left(\mathrm{m}_1 \boldsymbol{u}\right) & + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_1 \boldsymbol{u} \otimes \boldsymbol{u}) & = & \mathrm{G}_{\mathrm{m}_1 \boldsymbol{u}}, \\ \partial_t \left(\mathrm{m}_1 e_d\right) & + & \partial_{\boldsymbol{x}} \cdot (\mathrm{m}_1 \boldsymbol{u} e_d) & = & \mathrm{G}_{\mathrm{m}_1 e_d}, \end{array} \right. \tag{A.4}$$

where the source terms are expressed as follows:

$$
\begin{array}{lcl}
G_{\mathrm{m}_{k/2}} & = & -\int_0^1 k/2 S^{k/2-1} \mathrm{R}_d(S) n(S) \mathrm{d}S, \\
G_{\mathrm{m}_1 \boldsymbol{u}} & = & -\boldsymbol{u} \int_0^1 \mathrm{R}_d(S) n(S) \mathrm{d}S + \int_0^1 S \boldsymbol{F}_d(S) n(S) \mathrm{d}S, \\
G_{\mathrm{m}_1 e_d} & = & -e_d \int_0^1 \mathrm{R}_d(S) n(S) \mathrm{d}S + C_{v,l}(T_d) \int_0^1 S e_d^*(S) n(S) \mathrm{d}S.
\end{array}
\tag{A.5}
$$

The closure of the system is achieved by a continuous reconstruction through the maximization of Shannon Entropy.

We use the operator splitting technique to solve numerically the system (A.4). The transport part can be solved in the same way as it was done in chapter 7 of the paper. In the following, we focus on the resolution of the source term part. Let us consider a spatial homogeneous domain, where only source terms are involved, the equation system becomes:

$$
d_t \mathcal{M} = \mathcal{S}(t, \mathcal{M}) \tag{A.6}
$$

where $\mathcal{M} = (\mathrm{m}_0, \mathrm{m}_{1/2}, \mathrm{m}_1, \mathrm{m}_{3/2}, \mathrm{m}_1 \boldsymbol{u}, \mathrm{m}_1 e_d)^T$ is the unknown vector and $\mathcal{S} = (-\mathrm{R}_d n^{ME}(\mathbf{m})|_{S=0}, G_{\mathrm{m}_0}, G_{\mathrm{m}_{1/2}}, G_{\mathrm{m}_1}, G_{\mathrm{m}_{3/2}}, G_{\mathrm{m}_1 \boldsymbol{u}}, G_{\mathrm{m}_1 e_d})^T$ is the source term. The four moments are computed by using **NEMO** algorithm with a slight adaptation in the fourth step, where equation (8.26) is replaced by:

$$
\mathrm{m}_{k/2}(t_n + dt) = \sum_{j=1}^{n_q} w_j \widetilde{\phi}(t_n + dt; t_n, S_j)^{k/2}, \tag{A.7}
$$

where the weight w_j and abscissas S_j are determined in the third step of the algorithm and the $\widetilde{\phi}(t; t_0, S_o)$ is defined as follows:

$$
\left\{
\begin{array}{lcl}
\dfrac{d\widetilde{\phi}(t; t_0, S_o)}{dt} & = & \mathrm{R}_S(t, \widetilde{\phi}(t; t_0, S_o), \boldsymbol{u}, T_d), \\
\widetilde{\phi}(t_0; t_0, S_o) & = & S_o,
\end{array}
\right.
\tag{A.8}
$$

where the temperature T_d can be determined from the averaged internal energy e_d.

The updated velocity and the internal energy are computed using the CQMOM technique, as explained in the section 8.3 of the paper. For $t \in [t_n, t_{n+1}]$, we write after subtracting the disappearance flux from the moments:

$$
\begin{array}{lcl}
(\mathrm{m}_1 \boldsymbol{u})^{[\widetilde{\phi}(t_n; t_{n+1}, 0), 1]}(t) & = & \sum_{i=1}^{n_q} w_i \widetilde{\phi}(t; t_n, S_i) \boldsymbol{c}_i(t) \\
(\mathrm{m}_1 e_d)^{[\widetilde{\phi}(t_n; t_{n+1}, 0), 1]}(t) & = & \sum_{i=1}^{n_q} w_i \widetilde{\phi}(t; t_n, S_i) e_i(t)
\end{array}
\tag{A.9}
$$

with

$$
\left\{
\begin{array}{lcl}
\dfrac{d\boldsymbol{c}_i}{dt} & = & \boldsymbol{F}(t, \widetilde{\phi}(t; t_n, S_i), \boldsymbol{c}_i) \\
\boldsymbol{c}_i(t = t_n) & = & \boldsymbol{u}(t_n),
\end{array}
\right.
\tag{A.10}
$$

and

$$\left\{\begin{array}{rcl} \dfrac{de_i}{dt} & = & \boldsymbol{e}^*(t, \widetilde{\phi}(t; t_n, S_i), \boldsymbol{c}_i, T_d) \\ e_i(t = t_n) & = & e_d(t_n), \end{array}\right. \tag{A.11}$$

We recall that the superscript $[\widetilde{\phi}(t_n; t_{n+1}, 0), 1]$ refers to the moments of the measure defined in the support $[\widetilde{\phi}(t_n; t_{n+1}, 0), 1]$, where $\widetilde{\phi}(t_n; t_{n+1}, 0)$ is the size of the last evaporated droplet.

Appendix B

Evolution equation for the area density measure

In this section, let us derive the evolution equation for the area density measure δ_I. First, let us recall that δ_I is defined as a distribution function by:

$$\delta_I(t,\boldsymbol{x}) = ||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\,\delta(g(t,\boldsymbol{x})).$$

The Lagrangian derivative $\dot{\bullet}$ of δ_I reads:

$$\begin{aligned}\dot{\delta}_I(t,\boldsymbol{x}) &= \partial_t\delta_I + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(\delta_I)\\ &= \left[\partial_t\left(||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right) + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right)\right)\right]\,\delta(g(t,\boldsymbol{x}))\\ &\quad + \left[\partial_t\left(\delta(g(t,\boldsymbol{x}))\right) + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\delta(g(t,\boldsymbol{x}))\right)\right)\right]\,||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\end{aligned}$$

The second term in the right-hand side of the previous equality is null, because of the equation on $g(t,\boldsymbol{x})$ which is:

$$\partial_t g + \mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(g) = 0.$$

Let us now compute the first term. On a one hand, we have:

$$\begin{aligned}\partial_t\left(||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right) &= \frac{\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))}{||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||} \cdot \partial_t\left(\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))\right)\\ &= \boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\partial_t g(t,\boldsymbol{x})\right)\right)\\ &= -\boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))\right)\right)\\ &= -\boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\mathrm{v}_I||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right)\right).\end{aligned}$$

On the other hand, one computes:

$$\begin{aligned}\mathbf{v}_I \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right)\right) &= \mathrm{v}_I\boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right)\right)\\ &= \boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\mathrm{v}_I||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\right)\right) - ||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\mathrm{v}_I\right)\right)\end{aligned}$$

And finally, one can find that:

$$\begin{aligned}\dot{\delta}_I(t,\boldsymbol{x}) &= -||\boldsymbol{\nabla}_{\boldsymbol{x}}(g(t,\boldsymbol{x}))||\boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\mathrm{v}_I\right)\right)\,\delta(g(t,\boldsymbol{x}))\\ &= -\delta_I\,\boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}\left(\left(\mathrm{v}_I\right)\right).\end{aligned} \tag{B.1}$$

Equation (B.1) can be further developed, by noting that:

$$-\delta_I\, \boldsymbol{n} \cdot \boldsymbol{\nabla}_{\boldsymbol{x}}((\mathrm{v}_I)) = -\delta_I\, \left[\nabla_{\boldsymbol{x}} \cdot \mathbf{v}_I - \mathrm{v}_I \nabla_{\boldsymbol{x}} \cdot \boldsymbol{n}\right]$$

and

$$\nabla_{\boldsymbol{x}} \cdot \boldsymbol{n} = 2H.$$

Finally, one has:

$$\dot{\delta}_I(t, \boldsymbol{x}) = -\delta_I\, \nabla_{\boldsymbol{x}} \cdot \mathbf{v}_I + 2H\delta_I \mathrm{v}_I. \tag{B.2}$$

Appendix C

Taking into account the degrees of freedom of the DG discretization

In this section, we reuse the same notation as in section 7.3. The objective is to present the resolution of source terms in the framework of the Runge Kutta Discontinuous Galerkin (RKDG) discretization. We recall that the solution in a cell m is written as follows:

$$\boldsymbol{U}_h(t, \boldsymbol{x}) = \sum_{i=1}^{(k+1)^d} \boldsymbol{W}_i^m(t) \phi_i(\Xi_m(\boldsymbol{x})), \tag{C.1}$$

where d is the space dimension. In chapter 7, the method is presented for $d = 2$, but its extension to other dimension is straightforward. We use, as in the last chapter, a Lagrangian polynomial basis associated to the abscissa of $(k+1)^2$ Gauss Legendre quadrature points in the domain $[-1/2, 1/2]^d$. We denote simply by $(\boldsymbol{x}_q)_{1 \leq q \leq (k+1)^d}$ the sequence of abscissa points of Gauss Legendre quadrature in the domain of the cell m. We have then for all $1 \leq q \leq (k+1)^d$:

$$\boldsymbol{U}_h(t, \boldsymbol{x}_q) = \boldsymbol{W}_q^m(t). \tag{C.2}$$

Considering only the source terms part of the system of equations, we can show that each degree of freedom $\boldsymbol{W}_q^m(t)$ satisfies the ODE system (8.29) and its evolution can be solved using the method described in section 8.3. However, we need to ensure that the solution at Gauss-Legendre points is realizable, i.e. the first four components of $\boldsymbol{W}_q^m(t)[1:4] = (\mathrm{m}_0^{m,q}(t), \dots, \mathrm{m}_{3/2}^{m,q}(t))^t$ represent real fractional moments. Indeed, the limitation procedure of Zhang et al.(2012) ensures a realizable solution only at the Gauss-Legendre-Lobatto (GLL) points and for the mean solution in the cell. Our solution consists in using the same limitation procedure but this time at the Gauss-Legendre points instead of the GLL points. Thus, we obtain a realizable solution for the degree of freedoms, which can be updated using the source terms solver presented in the section 8.3. The final resolution with transport RKDG transport scheme can be summarized as follows:

- We solve the ODE system (7.36), using a Runge-Kutta method, to compute the evolution of degrees of freedom due to the transport in the physical space.
- We apply the limitation procedure to ensure the realizability at Gauss-Legendre points [1].
- We solve the source terms ODE (8.29) for each degree of freedom.

---

[1] The coefficient θ in (7.46) is computed here according to Gauss-Legendre points instead of Gauss-Legendre-Lobbato points.

- We apply the limitation procedure given in (7.47) to ensure the realizability of the moments and the maximum principle on the velocity at GLL points.

We note that we can alternate the resolution of the transport part and the source terms to preserve high order resolution.

References

E. Abbena, S. Salamon, and A. Gray. *Modern Differential Geometry of Curves and Surfaces with MATHEMATICA*. CRC Press, 2006. *(p. 77)*

B. Abramzon and W. A. Sirignano. Droplet vaporisation model for spray combustion calculations. *International Journal of Heat and Mass Transfer*, 9:1605–1618, 1989. *(p. 37)*

G. Agbaglah, R. Chiodi, and O. Desjardins. Numerical simulation of the initial destabilization of an air-blasted liquid layer. *Journal of Fluid Mechanics*, 812:1024–1038, 2017. *(p. 6, 19)*

G. Allaire, S. Clerc, and S. Kokh. A five equation model for the simulation of interfaces between compressible fluids. *J Comp Phys*, 181(2):577–616, 2002. *(p. 9)*

A. Ambroso, C. Chalons, F. Coquel, and T. Galié. Relaxation and numerical approximation of a two-fluid two-pressure diphasic model. *ESAIM: Mathematical Modelling and Numerical Analysis*, 43(6):1063–1097, 2009. *(p. 24)*

S. V. Apte, K. Mahesh, P. Moin, and J. C. Oefelein. Large-eddy simulation of swirling particle-laden flows in a coaxial-jet combustor. *IJMF*, 29(8):1311–1331, 2003. *(p. 32)*

N. Ashgriz and J. Poo. Coalescence and separation in binary collisions of liquid drops. *J. Fluid Mech*, 221:183 – 204, 1990. *(p. 90)*

C. Augonnet, S. Thibault, R. Namyst, and P.-A. Wacrenier. StarPU: a unified platform for task scheduling on heterogeneous multicore architectures. *Concurrency and Computation: Practice and Experience*, 23(2):187–198, 2011. *(p. 13, 191)*

M. R. Baer and J. W. 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. *(p. 23, 24)*

T. Barberon and P. Helluy. Finite volume simulation of cavitating flows. *Computers & Fluids*, 34(7):832 – 858, 2005. *(p. 24)*

M. J. Berger and J. Oliger. Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics*, 53(3):484–512, 1984. *(p. 162, 210)*

A. E. Bergles, J. G. Collier, J. M. Delhaye, G. F. Hewitt, and F. Mayinger. *Two-Phase Flow and Heat Transfer in the Power and Process Industries*. Hemisphere Publishing Corporation, New York, 1981. *(p. 23)*

A. Bernard-Champmartin and F. De Vuyst. A low diffusive lagrange-remap scheme for the simulation of violent air–water free-surface flows. *Journal of Computational Physics*, 274:19–49, 2014. *(p. 9)*

G. A. Bird. Molecular gas dynamics and the direct simulation of gas flows. *Oxford Science Publications*, 42, 1994. *(p. 8)*

J. Bohbot, O. Colin, A. Velghe, J.-B. Michel, M. Wang, P. K. Senecal, and E. Pomraning. An innovative approach combining adaptive mesh refinement, the ecfm3z turbulent combustion model, and the tki tabulated auto-ignition model for diesel engine cfd simulations. In *SAE Technical Paper*. 2016. *(p. 4)*

J. Bohbot, N. Gillet, and A. Benkenida. IFP-C3D: an unstructured parallel solver for reactive compressible gas flow with spray. *Oil & Gas Science and Technology*, 64(3):309–335, 2009. *(p. 5)*

M. Boileau, J. Lagarde, V. Dupif, F. Laurent, and M. Massot. Two-size moment Eulerian multi-fluid method describing the statistical trajectory crossing: modeling and numerical scheme. In *9th International Conference on Multiphase Flow*, Proceedings of the 9th International Conference on Multiphase Flow. Firenze, Italy, 2016. *(p. 46)*

F. Bouchut, S. Jin, and X. Li. Numerical approximations of pressureless and isothermal gas dynamics. *SIAM J. Num. Anal.*, 41:135–158, 2003. *(p. 12, 122, 124)*

J. A. Boure. Wave phenomena and one-dimensional two-phase flow models. *Multiphase Science and Technology*, 9(1):37–62, 1997. *(p. 26)*

J. Brackbill, D. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100(2):335 – 354, 1992. *(p. 102)*

C. Burstedde, L. C. Wilcox, and O. Ghattas. `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing*, 33(3):1103–1133, 2011. *(p. xiv, 10, 13, 156, 162, 164, 165, 182)*

S. Candel. *Mécabique des fluides.* Dunod, Paris, 1990. *(p. 18, 32)*

S. Candel and T. Poinsot. Flame stretch and the balance equation for the flame area. *Combustion Science and Technology*, 70:1–15, 1990. *(p. 29)*

R. Canu, C. Dumouchel, B. Duret, M. Essadki, M. Massot, T. Ménard, S. Puggelli, J. Reveillon, and F. Demoulin. Where does the drop size distribution come from? *ILASS Conference paper*, pages 1–8, 2017. *(p. 84, 93, 98, 110)*

S. de Chaisemartin. *Eulerian models and numerical simulation of turbulent dispersion for polydisperse evaporation sprays.* Ph.D. thesis, Ecole Centrale Paris, France, 2009. Available on TEL : `http://tel.archives-ouvertes.fr/tel-00443982/en/`. *(p. 5, 12, 37, 45, 124, 156, 177, 208)*

S. de Chaisemartin, L. Fréret, D. Kah, F. Laurent, R. Fox, J. Reveillon, and M. Massot. Eulerian models for turbulent spray combustion with polydispersity and droplet crossing. *Comptes Rendus Mécanique*, 337:438–448, 2009. Special Issue 'Combustion for Aerospace Propulsion'. *(p. 8, 80, 121, 122)*

G. Chanteperdrix, P. Villedieu, and J. Vila. A compressible model for separated two-phase flows computations. In *ASME Fluids Engineering Division Summer Meeting*. Montreal, 2002. *(p. 9)*

R. Clift, J. R. Grace, and M. E. Weber. *Bubbles, Drop and Particles.* Academic Press, New York, 1978. *(p. 36)*

B. Cockburn, S. Hou, and C.-W. Shu. The runge-kutta discontinuous galerkin method for conservation laws v. *Mathematics of Computation*, 54:545–581, 1990. *(p. 130)*

B. Cockburn and C.-W. Shu. The runge-kutta discontinuous galerkin method for conservation laws v. *Journal of Computational Physics*, 141(2):199–224, 1998. *(p. 122, 208)*

O. Colin and A. Benkenida. The 3-Zones Extended Coherent Flame Model (ECFM3Z) for Computing Premixed/Diffusion Combustion. *Oil & Gas Science and Technology*, 59(6):593–609, 2004. *(p. 4)*

O. Colin, A. Benkenida, and C. Angelberger. Ignition and Combustion Phenomena in Highly Stratified Gasoline Engines. *Oil & Gas Science and Technology*, 58(1):47–62, 2003. *(p. 4)*

P. Cordesse. *Study of combustion instabilities in liquid-propellant rocket engines: diffuse interface model and kinetic model coupling for primary atomization simulations.* Ph.D. thesis, Université Paris-Saclay, Polytechnique, 2019. *(p. 103, 116, 210)*

L. Cunha. *Simulation aux Grandes échelles de la Dynamique des Flammes Diphasiques* *en Ecoulement Tournant soumises à une modulation externe*. Ph.D. thesis, Université Paris-Saclay, CentraleSupélec, 2018. *(p. 32)*

J. M. Delhaye and J. A. Boure. *General equations and two-phase flow modeling, in Handbook of Multiphase Systems*. Academic Press, Inc., Boston, MA, 1982. *(p. 26)*

S. Descombes, M. Duarte, T. Dumont, F. Laurent, V. Louvet, and M. Massot. Analysis of operator splitting in the non-asymptotic regime for nonlinear reaction-diffusion equations. application to the dynamics of premixed flames. *SIAM J. Numerical Analysis*, 52:1311–1334, 2014. *(p. 106, 121)*

O. Desjardins, J. McCaslin, M. Owkes, and P. Brady. Direct numerical and large-eddy simulation of primary atomization in complex geometries. *Atomization and Sprays*, 23(11):1001–1048, 2013. *(p. 7)*

H. Dette and W. J. Studden. *The theory of canonical moments with applications in statistics, probability, and analysis*. John Wiley & Sons Inc., New York, 1997. *(p. 12, 54, 55, 61, 62)*

M. B. Devassy, C. Habchi, and E. Daniel. Atomization modelling of liquid jets using a two-surface-density approach. *Journal of Atomization and Sprays*, 25(1):47–80, 2015. *(p. 4, 10, 23, 28, 29, 102)*

R. Di Battista. *Eulerian unified modeling for separated and disperse two-phase flows based on a geometrical treatment of the interface*. Ph.D. thesis, Université Paris-Saclay, Polytechnique, 2020. *(p. 103, 116, 210)*

F. Doisneau. *Eulerian modeling and simulation of polydisperse moderately dense coalescing spray flows with nanometric-to-inertial droplets: application to Solid Rocket Motors*. Ph.D. thesis, Ecole Centrale Paris, 2013. Available at `http://tel.archives-ouvertes.fr/tel-00966185`. *(p. xi, 8, 43)*

F. Doisneau, F. Laurent, A. Murrone, J. Dupays, and M. Massot. Eulerian multi-fluid models for the simulation of dynamics and coalescence of particles in solid propellant combustion. *J. Comp. Phys.*, 234:230–262, 2013. *(p. 38)*

F. Doisneau, A. Sibra, J. Dupays, A. Murrone, F. Laurent, and M. Massot. Numerical strategy for unsteady two-way coupled polydisperse sprays: application to solid-rocket instabilities. *Journal of Propulsion and Power*, 3:727–748, 2014. *(p. 106, 121)*

D. Drew and S. Passman. *Theory of multicomponent fluids*, volume 135. Springer, 1999. *(p. 9, 19, 20, 22, 26, 38, 72)*

D. A. Drew. Mathematical modeling of two-phase flow. *Ann. Rev. Fluid. Mech.*, 15:261–91, 1983. *(p. 22)*

D. A. Drew. Evolution of geometric statistics. *SIAM J. Appl. Math.*, 50(3):649–666, 1990. *(p. 28, 53, 57, 58, 59, 70, 71, 73, 74, 208)*

F. Drui. *Eulerian modeling and simulations of separated and disperse two-phase flows: development of a unified modeling approach and associated numerical methods for highly parallel computations*. Ph.D. thesis, Université Paris-Saclay, CentraleSupélec, 2017. *(p. xi, xii, xiv, 4, 10, 13, 27, 95, 99, 103, 104, 107, 116, 161, 164, 167, 169, 170, 209)*

F. Drui, A. Fikl, P. Kestener, S. Kokh, A. Larat, V. Le Chenadec, and M. Massot. Experimenting with the p4est library for AMR simulations of two-phase flows. *ESAIM : Proceedings and Surveys*, 53:232–247, 2016a. *(p. 27, 156, 162, 167)*

F. Drui, S. Kokh, A. Larat, and M. Massot. A hierarchy of simple hyperbolic two-fluid models for bubbly flows. *submitted*, pages 1–23, 2016b. Available on HAL, `https://hal.archives-ouvertes.fr/hal-01349441`. *(p. 9, 23, 24, 57, 103, 167, 210)*

M. Duarte. *Adaptive numerical methods in time and space for the simulation of multiscale reaction fronts*. Ph.D. thesis, Ecole Centrale de Paris, 2011. Available on TEL `https://tel.archives-ouvertes.fr/tel-00667857`. *(p. 173, 210)*

A. Dubey, A. Almgren, J. Bell, M. Berzins, S. Brandt, G. Bryan, P. Colella, D. Graves, M. Lijewski, F. Löffler, et al. A survey of high level frameworks in block-structured adaptive mesh refinement packages. *Journal of Parallel and Distributed Computing*, 74(12):3217–3227, 2014. *(p. 162)*

G. Dufour. *Modélisation multi-fluide eulérienne pour les écoulements diphasiques à inclusions dispersées*. Ph.D. thesis, Université Toulouse III, 2005. *(p. 36, 39, 46)*

C. Dumouchel, T. Ménard, and W. Aniszewski. Towards an interpretation of the scale diffusivity in liquid atomization process: An experimental approach. *Physica A: Statistical Mechanics and its Applications*, 438(Supplement C):612 – 624, 2015. *(p. 98)*

V. Dupif. *Modeling and simulation of two-phase flows inside solid rocket motors through an eulerian approach with size and velocity polydispersion*. Ph.D. thesis, Paris-Saclay University, 2018. *(p. 42)*

O. Emre. *Modeling of spray polydispersion with two-way turbulent interactions for high pressure direct injection in engines*. Ph.D. thesis, Ecole Centrale Paris, 2014. Available on TEL `https://tel.archives-ouvertes.fr/tel-01089937`. *(p. 5)*

O. Emre, D. Kah, S. Jay, Q.-H. Tran, A. Velghe, S. De Chaisemartin, F. Laurent, and M. Massot. Eulerian Moment Methods for Automotive Sprays. *Atomization and Sprays*, 25:189–254, 2015. *(p. 8, 32)*

M. Essadki, S. de Chaisemartin, M. Massot, F. Laurent, A. Larat, and S. Jay. High order moment methods and adaptive mesh refinement for polydisperse evaporating spray simulations. *Oil & Gas Science and Technology*, 71:1–25, 2016. *(p. 156, 167)*

M. Essadki, J. Jung, A. Larat, M. Pelletier, and V. Perrier. A task-driven implementation of a simple numerical solver for hyperbolic conservation laws. *Submitted to ESAIM: PROCEEDINGS AND SURVEYS*, 2017. *(p. xv, 191, 194)*

G. Faccanoni, S. Kokh, and G. Allaire. Modelling and simulation of liquid-vapor phase transition in compressible flows based on thermodynamical equilibrium. *ESAIM: Mathematical Modelling and Numerical Analysis*, 46(5):1029–1054, 2012. *(p. 24)*

G. M. Faeth. Evaporation and combustion of spray. *Progress in Energy and Combustion-Science*, 9:1–76, 1983. *(p. 37)*

R. O. Fox and P. Vedula. Quadrature-based moment model for moderately dense polydisperse gas-particle flows. *Ind. Eng. Chem.*, 49:5174–5187, 2010. *(p. 39)*

D. Fuster, A. Bagu, T. Boeck, L. L. Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski. Simulation of primary atomization with an octree adaptive mesh refinement and VOF method. *International Journal of Multiphase Flow*, 35(1):550–565, 2009. *(p. 7)*

T. Gallouet, J.-M. Hérard, and N. Seguin. Numerical modeling of two-phase flows using the two-fluid two-pressure approach. *Mathematical Models and Methods in Applied Sciences*, 14(05):663–700, 2004. *(p. 24)*

M. García, E. Riber, O. Simonin, and T. Poinsot. Comparison between euler/euler and euler/lagrange les approaches for confined bluff-body gas-solid flow. In *International Conference on Multiphase Flow*, pages CD–ROM – S3–Fri–A–62. Liepzig, Germany, 2007. *(p. 7)*

S. Gavrilyuk and R. Saurel. Mathematical and numerical modeling of two-phase compressible flows with micro-inertia. *Journal of Computational Physics*, 175:326–360, 2002. *(p. 9, 24, 103, 210)*

S. Ghods and M. Herrmann. A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods. *Physica Scripta*, 2013(T155):014050, 2013. *(p. 6, 7)*

E. Godlewski and P.-A. Raviart. *Numerical approximation of hyperbolic systems of con-*

servation laws, volume 118. Springer-Verlag, New York, 1996. *(p. 26)*

G. A. E. Godsave. Studies of the combustion of drops in a fuel spray: the burning of single drops of fuel. In *Proceedings of the 4th Symp. (International) on Combustion, The Comb. Institute*, pages 818–830. Baltimore, 1953. *(p. 37)*

R. Gordon. Error bounds in equilibrium statistical mechanics. *Journal of Mathematical Physics*, 9:655–663, 1968. *(p. 56, 62, 107, 146, 150)*

S. Gottlieb, C.-W. Shu, and E. Tadmor. Strong stability-preserving high-order time discretization methods. *SIAM Review*, 43(1):89–112, 2001. *(p. 130, 131, 132)*

J. Greenberg, I. Silverman, and Y. Tambour. On the origin of spray sectional conservation equations. *Combustion and Flame*, 93:90–96, 1993. *(p. 8)*

V. Guillemaud. *Modélisation et simulation numérique des écoulements diphasiques par une approche bifluide à deux pressions.* Ph.D. thesis, 2007a. 2007AIX11005. *(p. 24)*

V. Guillemaud. Modelling and numerical simulation of strongly unbalanced two-phase flows. In *18th AIAA Computational Fluid Dynamics Conference.* 2007b. *(p. 24)*

H. Gzyl and A. Tagliani. Hausdorff moment problem and fractional moments. *Appl. Math. Comput.*, 216(11):3319–3328, 2010. *(p. 53)*

G. Hannebique, P. Sierra, E. Riber, and B. Cuenot. Large Eddy Simulation of reactive two-phase flow in an aeronautical multipoint burner. *Flow, Turbulence and Combustion*, accepted in September 2012, 2013. *(p. 8)*

F. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8(12):2182–2189, 1965. *(p. 6)*

A. Harten. Adaptive multiresolution schemes for shock computations. *Journal of Computational Physics*, 115:319–338, 1994. *(p. 210)*

A. Harten. Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Communication on Pure Applied Mathematics*, 48:1305–1342, 1995. *(p. 173, 174)*

M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics*, 227:2674–2706, 2008. *(p. 10)*

M. Herrmann. A parallel eulerian interface tracking/lagrangian point particle multi-scale coupling procedure. *Journal of Computational Physics*, 229:745–759, 2010. *(p. 10)*

C. Hirt, A. Amsden, and J. Cook. An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of Computational Physics*, 14(3):227–253, 1974. *(p. 6)*

C. Hirt and B. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics*, 39(1):201–225, 1981. *(p. 6)*

O. Hurisse. Numerical simulations of steady and unsteady two-phase flows using a homogeneous model. *Computers & Fluids*, 152(Supplement C):88 – 103, 2017. *(p. 24)*

J. Hylkema. *Modélisation cinétique et simulation numérique d'un brouillard dense de gouttelettes. Application aux propulseurs à poudre.* Ph.D. thesis, ENSAE, 1999. *(p. 39)*

P. N. Inverardi, G. Pontuale, A. Petri, and A. Tagliani. Hausdorff moment problem via fractional moments. *Appl. Math. Comput.*, 144(1):61–74, 2003. *(p. 53)*

M. Ishii. *Thermo-Fluid dynamic theory of two-phase flow.* Eyrolles, 1975. *(p. 9, 20)*

M. Ishii and T. Hibiki. *Thermo-Fluid Dynamics of Two-Phase Flow.* 2. Springer-Verlag New York, 2011. *(p. 26)*

D. Kah. *Taking into account polydispersity in the framework of a coupled Euler-Lagrange approach for the modeling of liquid fuel injection in internal combustion engines.* Ph.D. thesis, Ecole Centrale de Paris, 2010. Available online at `http://tel.archives-ouvertes.fr/tel-00618786/en/`. *(p. 5, 9, 11, 12, 54, 96, 207)*

D. Kah, O. Emre, Q. Tran, S. de Chaisemartin, S. Jay, F. Laurent, and M. Massot. High order moment method for polydisperse evaporating spray with mesh movement:

application to internal combustion engines. *Int. J. Multiphase Flows*, 71:38–65, 2015. *(p. 9, 32)*

D. Kah, F. Laurent, M. Massot, and S. Jay. A high order moment method simulating evaporation and advection of a polydisperse spray. *Journal of Computational Physics*, 231(2):394–422, 2012. *(p. 5, 8, 12, 46, 53, 121, 122, 124, 125, 126, 156, 208)*

M. Kang, R. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000. *(p. 84)*

J. N. Kapur and H. K. Kesavan. *Entropy optimization principles with applications.* Academic Press, Inc., Boston, MA, 1992. *(p. 54)*

I. Kataoka. Local instant formulation of two-phase flow. *International Journal of Multiphase Flow*, 12(5):745 – 758, 1986. *(p. 28)*

G. Kindlmann, R. Whitaker, T. Tasdizen, and T. Moller. Curvature-based transfer functions for direct volume rendering: Methods and applications. In *Proceedings of the 14th IEEE Visualization 2003 (VIS'03)*, pages 67–. IEEE Computer Society, Washington, DC, USA, 2003. *(p. 69, 70)*

S. Kokh and F. Lagoutière. An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model. *Journal of Computational Physics*, 229:2773–2809, 2010. *(p. 9, 17, 27)*

A. Larat, M. Massot, and A. Vié. A stable, robust and high order accurate numerical method for eulerian simulation of spray and particle transport on unstructured meshes. In *Proceedings of the summer program 2012, Center for Turbulence Research, Stanford University*, pages 1–10. 2012. *(p. 128)*

F. Laurent. *Modélisation mathématique et numérique de la combustion de brouillards de gouttes polydispersés.* Ph.D. thesis, Université Claude Bernard, Lyon 1, 2002. *(p. 38)*

F. Laurent. Numerical analysis of eulerian multi-fluid models in the context of kinetic formulations for dilute evaporating sprays. *Mathematical Modeling and Numerical Analysis*, 3:431–468, 2006. *(p. 46)*

F. Laurent, M. Massot, and P. Villedieu. Eulerian multi-fluid modeling for the numerical simulation of coalescence in polydisperse dense liquid sprays. *Journal of Computational Physics*, 194:505–543, 2004. *(p. 38, 45)*

F. Laurent, A. Sibra, and F. Doisneau. Two-size moment Eulerian multi-fluid model: a robust and high-fidelity description of polydisperse moderately dense evaporating sprays. *Comm. in Comput. Phys.*, 20:902–943, 2016. *(p. 8, 46)*

V. Le Chenadec and H. Pitsch. A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows. *J. Comput. Phys.*, 249:185–203, 2013. *(p. 7)*

S. Le Martelot. *Contribution á la simulation numérique directe de l'ébullition.* Ph.D. thesis, Aix Marseille Université, 2013. *(p. 3)*

S. Le Martelot, R. Saurel, and B. Nkonga. Towards the direct numerical simulation of nucleate boiling flows. *International Journal of Multiphase Flow*, 66:62–78, 2014. *(p. 3)*

C. Le Touze. *Coupling between separated and dispersed two-phase flow models for the simulation of primary atomization in cryogenic combustion.* Ph.D. thesis, Universite Nice Sophia Antipolis, 2015. Available on TEL `https://tel.archives-ouvertes.fr/tel-01250527`. *(p. 10)*

R. Lebas, T. Menard, P. A. Beau, A. Berlemont, and F. X. Demoulin. Numerical simulation of primary break-up and atomization: DNS and modelling study. *Int. J. Multiphase Flows*, 35(3):247–260, 2009. *(p. 6, 7, 10, 28, 29, 58, 112)*

D. Lhuillier, C.-H. Chang, and T. G. Theofanous. On the quest for a hyperbolic effective-field model of disperse flows. *Journal of Fluid Mechanics*, 731:184–194, 2013. *(p. 23)*

D. Marchisio and R. Fox. Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science*, 36:43–73, 2005. *(p. 8, 44)*

D. Marchisio, J. Pikturna, R. Fox, and R. Vigil. Quadrature method of moments for population-balance equations. *AIChE Journal*, 49(5):1266–1276, 2003. *(p. 42)*

L. Martinez, A. Benkenida, and B. Cuenot. A model for the injection boundary conditions in the context of 3D simulation of diesel spray: methodology and validation. *Fuel*, 89(1):219–228, 2010. *(p. 32)*

F. Mashayek. Direct numerical simulations of evaporating droplet dispersion in forced mach number turbulence. *Heat Mass Transfer*, 52:2601–2617, 1998. *(p. 7)*

M. Massot, F. Laurent, D. Kah, and S. de Chaisemartin. A robust moment method for evaluation of the disappearance rate of evaporating sprays. *SIAM J. Appl. Math.*, 70:3203–3234, 2010. *(p. 5, 8, 13, 41, 46, 53, 54, 56, 57, 66, 143, 144, 145, 209)*

J.-P. Matas and A. Cartellier. Flapping instability of a liquid jet. *Comptes Rendus Mécanique*, 341(1):35 – 43, 2013. Combustion, spray and flow dynamics for aerospace propulsion. *(p. xi, 7)*

L. R. Mead and N. Papanicolaou. Maximum entropy in the problem of moments. *J. Math. Phys.*, 25(8):2404–2417, 1984. *(p. 8, 48, 53, 56, 63, 65)*

T. Menard, S. Tanguy, and A. Berlemont. Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, 33(5):510–524, 2007. *(p. 6, 12, 19, 67, 84, 110)*

M. Meyer, M. Desbrun, P. Schroder, and A. H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. *Visualization and Mathematics III*, pages 35–57, 2001. *(p. 85)*

C. Morel. *Modélisation multidimensionnelle des écoulement diphasique gaz-liquid.* Ph.D. thesis, Ecole Centrale de Paris, 1997. *(p. 26, 28)*

C. Morel. *Mathematical Modeling of Disperse Two-Phase Flows.* Springer International Publishing, 2015. *(p. 28, 29, 70)*

S. Muller. *Adaptive Multiscale Schemes for Conservation Laws.* Springer, 2003. *(p. 173)*

A. Murrone and H. Guillard. A five equation reduced model for compressible two phase flow problems. *Journal of Computational Physics*, 205:664–698, 2005. *(p. 9)*

M. Ndjinga. *Quelques aspects de modélisation et dánalyse des sytèmes issus des écoulements diphasique.* Ph.D. thesis, Centrale Paris, 2007. *(p. 25)*

T. T. Nguyen, F. Laurent, R. O. Fox, and M. Massot. Solution of population balance equations in applications with fine particles: mathematical modeling and numerical schemes. *J. Comp. Phys.*, 325:129–156, 2016. *(p. 8)*

P. J. O'Rourke. *Collective drop effects on vaporizing liquid sprays.* Ph.D. thesis, Princeton University, 1981. *(p. 36)*

P. J. O'Rourke and A. Amsden. The tab method for numerical calculation of spray droplet breakup. Technical Report 87545, Los Alamos National Laboratory, Los Alamos, New Mexico, 1987. *(p. 39)*

G. Pianet, S. Vincent, J. Leboi, J. P. Caltagirone, and M. Anderhuber. Simulating compressible gas bubbles with a smooth volume tracking 1-fluid method. *International Journal of Multiphase Flow*, 36:273–283, 2010. *(p. 6, 19)*

T. Poinsot and D. Veynante. Theoretical and numerical combustion - Second Edition. 2005. *(p. 32, 34)*

S. B. Pope. The evolution of surfaces in turbulence. *Journal of Engineering Science*, 6(5):445–469, 1988. *(p. 12, 67, 68, 71, 72)*

W. Reed and T. Hill. *Triangular mesh methods for the neutron transport equation.* 1973. *(p. 127)*

R. D. Reitz. Mechanisms of atomization processes in high-pressure vaporizing sprays. *Atomization and Spray Technology*, 3:309–337, 1987. *(p. 105)*

J. Reveillon and F. Demoulin. Effects of the preferential segregation of droplets on evaporation and turbulent mixing. *Journal of Fluid Mechanics*, 583:273–302, 2007. *(p. 185)*

M. Sabat. *Eulerian modeling and numerical methods for the description of turbulent polydisperse sprays.* Ph.D. thesis, Université Paris-Saclay, CentraleSupélec, 2016. *(p. 44, 187)*

M. Sabat, A. Larat, A. Vié, and M. Massot. On the development of high order realizable schemes for the Eulerian simulation of disperse phase flows: a convex-state preserving Discontinuous Galerkin method. *Journal of Computational Multiphase Flows*, 6(3):247–270, 2014. *(p. 185)*

M. Sanjosé, J.-M. Senoner, F. Jaegle, B. Cuenot, S. Moreau, and T. Poinsot. Fuel injection model for Euler-Euler and Euler-Lagrange Large-Eddy Simulations of an evaporating spray inside an aeronautical combustor. *International Journal of Multiphase Flow*, 37(5):514–529, 2011. *(p. 32)*

R. Saurel and R. Abgrall. A multiphase godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics*, 150(2):425 – 467, 1999. *(p. 4, 22, 24, 26, 27)*

R. Saurel and O. L. Metayer. A multiphase model for compressible flows with interfaces, shocks, detonation waves and cavitation. *Journal of Fluid Mechanic*, 431:239–271, 2000. *(p. 23, 26)*

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*, 228:1678–1712, 2009. *(p. 24)*

N. Seguin. *Modélisation et simulation numérique des écoulements diphasiques.* Theses, Université de Provence - Aix-Marseille I, 2002. Available on: https://tel.archives-ouvertes.fr/tel-00003139/file/tel-000031391.pdf. *(p. 24)*

R. K. Shukla, C. Pantano, and J. B. Freund. An interface capturing method for the simulation of multi-phase compressible flows. *Journal of Computational Physics*, 229:7411–7439, 2010. *(p. 17, 27)*

S. Sibendu, I. R. Anita, E. L. Douglas, and K. A. Suresh. Effect of nozzle orifice geometry on spray, combustion, and emission characteristics under diesel engine conditions. *Fuel*, 90(3), 2011. *(p. 3)*

A. Sibra. *Eulerian Multi-Fluid modeling and simulation of evaporation and combustion of polydisperse sprays in solid rocket motors.* Theses, Université Paris-Saclay, 2015. *(p. 8)*

W. Sirignano. *Fluid dynamics and transport of droplets and sprays.* Cambridge University Press, New York, 1999. *(p. 36, 37)*

K. So, X. Hu, and N. Adams. Anti-diffusion interface sharpening technique for two-phase compressible flow simulations. *Journal of Computational Physics*, 231:4304–4323, 2012. *(p. 17, 27)*

D. B. Spalding. The combustion of liquid fuels. In *Proceedings of the 4th Symp. (International) on Combustion, The Comb. Institute*, pages 847–864. Baltimore, 1953. *(p. 37)*

G. Talenti. Recovering a function from a finite number of moments. *Inverse Problems*, 3(3):501–517, 1987. *(p. 53)*

L. Thomas, L. Hélio, V. Antonio Wilson, and T. Geovan. Efficient implementation of marching cubes cases with topological guarantees. *Journal of Graphics Tools*, 8(2):1–15, 2003. *(p. 84)*

A. Tiwari, J. B. Freund, and P. Carlos. A diffuse interface model with immiscibility preservations. *Journal of Computational Physics*, 231:4304–4323, 2013. *(p. 17, 27)*

B. Truchot. *Développement et validation d'un modèle eulérien en vue de la simulation des jets de carburants dans les moteurs à combustion interne*. Ph.D. thesis, Institut National Polytechnique de Toulouse, 2005. *(p. 4)*

S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25 – 37, 1992. *(p. 6)*

A. Vallet and R. Borghi. Modélisation eulerienne de l'atomisation d'un jet liquide. *Comptes Rendus de l'Académie des Sciences - Series IIB - Mechanics-Physics-Astronomy*, 327(10):1015 – 1020, 1999. *(p. 10)*

A. Vallet, A. Burluka, and R. Borghi. Development of a eulerian model for the "atomization" of a liquid jet. *Atomization and Sprays*, 11:619–642, 2001. *(p. 10, 29, 58)*

G. Vaudor, T. Ménard, W. Aniszewski, M. Doring, and A. Berlemont. A consistent mass and momentum flux computation method for two phase flows. application to atomization process. *Computers & Fluids*, 152(Supplement C):204 – 216, 2017. *(p. xi, 6, 7, 12, 110)*

C. Vessiller. *Contribution à l'étude des brouillards denses et dilués par la simulation numérique Euler/Euler et Euler/Lagrange*. Ph.D. thesis, Ecole Centrale Paris, 2007. *(p. 4)*

A. Vié, F. Doisneau, and M. Massot. On the Anisotropic Gaussian closure for the prediction of inertial-particle laden flows. *Comm. in Comput. Phys.*, 17(1):1–46, 2015a. *(p. 42, 44)*

A. Vié, B. Franzelli, Y. Gao, T. Lu, and M. Ihme. Analysis of segregation and bifurcation in turbulent spray flames: a 3d counterflow configuration. *Proceedings of the Combustion Institute*, 35:1675–1683, 2015b. *(p. 32)*

A. Vié, S. Jay, B. Cuenot, and M. Massot. Accounting for Polydispersion in the Eulerian Large Eddy Simulation of an Aeronautical-type Configuration. *Flow Turbulence and Combustion*, 90(3):545–581, 2013a. *(p. xi, 8, 31)*

A. Vié, F. Laurent, and M. Massot. Size-velocity correlations in high order moment methods for polydisperse evaporating sprays: modeling and numerical issues. *J. Comp. Phys.*, 237:277–310, 2013b. *(p. 5, 8, 42, 49, 57, 132, 151, 156)*

A. Vié, M. Sanjosé, S. Jay, B. Cuenot, and M. Massot. Evaluation of a multifluid mesoscopic eulerian formalism on the large eddy simulation of an aeronautical-type configuration. In *International Conference on Multiphase Flow 2010*. 2010. *(p. 5)*

P. Villedieu and J. Hylkema. Modèles numériques lagrangiens pour la phase dispersée dans les propulseurs à poudre. Rapport Final 1/3784.00/DTIM, ONERA, 2000. *(p. 39)*

S. Van der Walt, J. L. Schönberger, J. Nunez-Iglesias, F. Boulogne, J. D. Warner, N. Yager, E. Gouillart, and T. a. Yu. scikit-image: image processing in python. *PeerJ*, 2:e453, 2014. *(p. 84)*

F. Williams. Spray combustion and atomization. *Physics of Fluids*, 1:541–545, 1958. *(p. 8, 32, 36)*

C. Yuan, F. Laurent, and R. Fox. An extended quadrature method of moments for population balance equations. *Atomization and Sprays*, 51(1):1–23, 2012. *(p. 8)*

R. Zamansky, F. Coletti, M. Massot, and A. Mani. Turbulent thermal convection driven by heated inertial particles. *Journal of Fluid Mechanics*, 809:390–4437, 2016. *(p. 7)*

X. Zhang and C.-W. Shu. On maximum-principle-satisfying high order schemes for scalar conservation laws. *J. Comput. Phys.*, 229(9):3091–3120, 2010. *(p. 128)*

X. Zhang and C.-W. Shu. Positivity-preserving high order discontinuous galerkin schemes for compressible euler equations with source terms. *J. Comput. Phys.*, 230(4):1238–

1248, 2011. *(p. 122, 128, 132)*

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. *J. Sci. Comput.*, 50(1):29–62, 2012. *(p. 122, 128, 131, 195, 208, 217)*

H. Zhu, Z. Zhou, R. Yang, and A. Yu. Discrete particle simulation of particulate systems: Theoretical developments. *Chemical Engineering Science*, 62:3378–3396, 2007. *(p. 7)*

D. Zuzio and J. Estivalezes. An efficient block parallel amr method for two phase interfacial flow simulations. *Computers & Fluids*, 44(1):339 – 357, 2011. *(p. 10)*

universite PARIS-SACLAY

École doctorale de mathématiques Hadamard (EDMH)

Titre : Contribution à la modélisation eulérienne unifiée de l'injection : de la zone dense au spray polydispersé.

Mots Clefs : Écoulements diphasiques; spray polydisperse; phases séparée; méthodes numeriques d'order élevée; géometrie d'interface; modèles bifluides; schémas cinétiques; realizabilité; Calcul Haute Performance.

Résumé :

L'injection directe à haute pression du carburant dans les moteurs à combustion interne permet une atomisation compacte et efficace. Dans ce contexte, la simulation numérique de l'injection est devenue un outil fondamental pour la conception industrielle. Cependant, l'écoulement du carburant liquide dans une chambre occupée initialement par l'air est un écoulement diphasique très complexe ; elle implique une très large gamme d'échelles. L'objectif de cette thèse est d'apporter de nouveaux éléments de modélisation et de simulation afin d'envisager une simulation prédictive de ce type d'écoulement avec un coût de calcul abordable dans un contexte industriel. En effet, au vu du coût de calcul prohibitif de la simulation directe de l'ensemble des échelles spatiales et temporelles, nous devons concevoir une gamme de modèles d'ordre réduit prédictifs. En outre, des méthodes numériques robustes, précises et adaptées au calcul de haute performance sont primordiales pour des simulations complexes.

Cette thèse est dédiée au développement d'un modèle d'ordre réduit Eulérien capable de capter tant la polydispersion d'un brouillard de goutte dans la zone dispersée, que la dynamique de l'interface dans le régime de phases séparées. En s'appuyant sur une extension des méthodes de moments d'ordre élevé à des moments fractionnaires qui représentent des quantités géométriques de l'interface, et sur l'utilisation de variables géométriques en sous-échelle dans la zone où l'interface gaz-liquide ne peut plus être complètement résolue, nous proposons une approche unifiée où un ensemble de variables géométriques sont transportées et valides dans les deux régimes d'écoulement [...].

Title: Contribution to a unified Eulerian modeling of fuel injection: from dense liquid to polydisperse spray

Keywords : Two-phase flows; polydisperse sprays; separated phases; high order moment methods; interface geometry; two-fluid models; kinetic-based numerical schemes; realizability; HPC.

Abstract:

Direct fuel injection systems are widely used in combustion engines to better atomize and mix the fuel with the air. The design of new and efficient injectors needs to be assisted with predictive simulations. The fuel injection process involves different two-phase flow regimes that imply a large range of scales. In the context of this PhD, two areas of the flow are formally distinguished: the dense liquid core called separated phases and the polydisperse spray obtained after the atomization. The main challenge consists in simulating the combination of these regimes with an acceptable computational cost. Direct Numerical Simulations, where all the scales need to be solved, lead to a high computational cost for industrial applications. Therefore, modeling is necessary to develop a reduced order model that can describe all regimes of the flow. This also requires major breakthrough in terms of numerical methods and High Performance Computing (HPC).

This PhD investigates Eulerian reduced order models to describe the polydispersion in the disperse phase and the gas-liquid interface in the separated phases. First, we rely on the moment method to model the polydispersion in the downstream region of the flow. Then, we propose a new description of the interface by using geometrical variables. These variables can provide complementary information on the interface geometry with respect to a two-fluid model to simulate the primary atomization. The major contribution of this work consists in using a unified set of variables to describe the two regions: disperse and separated phases. In the case of spherical droplets, we show that this new geometrical approach can degenerate to a moment model similar to Eulerian Multi-Size Model (EMSM). However, the new model involves fractional moments, which require some specific treatments. This model has the same capacity to describe the polydispersion as the previous Eulerian moment models: the EMSM and the multi-fluid model. But, it also enables a geometrical description of the interface [...].

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France