Modèles eulériens et simulation numérique de la dispersion turbulente de brouillards qui s'évaporent

Stéphane De de Chaisemartin Chaisemartin

École Centrale Paris

THÈSE

présentée par

Stéphane DE CHAISEMARTIN

pour l'obtention du

GRADE de DOCTEUR

Formation doctorale : **Mathématiques Appliquées - Energétique**

Laboratoire d'accueil : **Laboratoire d'Energétique Moléculaire et Macroscopique, Combustion (EM2C) du CNRS et de l'ECP**

Modèles eulériens et simulation numérique de la dispersion turbulente de brouillards qui s'évaporent

Soutenue le 20 mars 2009

Composition du jury :

MM.	GIOVANGIGLI	V.	Président
	DESPRÉS	B.	Rapporteur
	POINSOT	T.	Rapporteur
	FOX	R. O.	Examinateur
Mme.	LAURENT	F.	Directrice de thèse
MM.	MASSOT	M.	Directeur de thèse
	RÉVEILLON	J.	Co-encadrant
	DUPAYS	J-M.	Invité
	LEDERLIN	T.	Invité
	WEY	S.	Invité

Ecole Centrale des Arts et Manufactures
Grand Etablissement sous tutelle
du Ministère de l'Education Nationale
Grande Voie des Vignes
92295 CHATENAY MALABRY Cedex
Tél. : 33 (1) 41 13 10 00 (standard)
Télex : 634 991 F EC PARIS

Laboratoire d'Energétique Moléculaire et Macroscopique, Combustion (E.M2.C.)
UPR 288, CNRS et Ecole Centrale Paris
Tel. : 33 (1) 41 13 10 31
Telecopie : 33 (1) 47 02 80 35

2009 - 2009ECAP0011

À Verónica
À Sarah
À Nathanaël
...
À la mémoire de Guillaume.

Remerciements

Un seul nom apparaît, et de manière triomphale, en page de garde de ce manuscrit. Si cela devait faire penser que ce travail n'est le fait que d'une seule personne, il serait nécessaire de déroger à l'usage et d'apposer la mention *et al.* derrière ce nom. Ces quelques lignes ont donc pour vocation d'essayer de mettre en lumière toutes les personnes qui se sont retrouvées, de près ou de loin, de plein gré ou contraint, embarquées dans cette aventure doctorale, et de leur exprimer toute ma reconnaissance.

Une thèse c'est bien sûr avant tout un sujet. Pour ma part, cela a commencé par un projet de 3$^{\text{ème}}$ année de l'ECP. J'ai été initié au cours de ce projet à l'analyse numérique et au calcul appliqués à la mécanique des fluides. Ma chance fut d'être alors guidé par ceux qui restent pour moi une référence : Marc Massot et Frédérique Laurent. De cette première étape initiatique, je retiendrai surtout, au delà de leur excellence scientifique, leur sens aigu de la pédagogie, qui m'a permis de rentrer progressivement dans cet univers complexe de la modélisation et de la simulation diphasique, et de me construire une motivation à toute épreuve. Je salue et remercie leur patience, mise à l'épreuve chaque semaine au cours des réunions de projets, par les incompréhensions et les erreurs de mes timides débuts.
Ce projet s'est donc transformé en sujet de thèse dont l'intitulé peut laisser rêveur. Un sujet audatieux, à la croisée de plusieurs disciplines : mathématiques appliquées, modélisation et calcul scientifique. Je tiens ici à remercier Frédérique et Marc qui ont mis en place un projet ambitieux : traiter des problèmes de sciences de l'ingénieur avec la rigueur des mathématiques appliquées et ce par un ingénieur...

Le doctorant ne se nourrissant pas seulement de sciences, une thèse c'est également un financement. A projet original, fincancement original, quasi triparti. Je remercie la DGA et le CNRS qui ont cofinancés ma bourse de docteur ingénieur. Plus précisément je remercie les départements MPPU pour les mathématiques appliquées et ST2I pour les sciences de l'ingénieur, qui ont tous les deux contribués au financement CNRS. Je m'excuse pour les sigles CNRS qui sont vraisemblablement caduques, ils étaient en vigueur à l'obtention du financement.
La thèse a également été soutenue par l'ACI Nouvelles Interfaces des Mathématiques "Modélisation des Milieux Réactifs : vers de nouveaux outils de simulation pour les problèmes multi-échelles", par le projet ANR jeunes chercheurs jéDYS, tout deux coordonnés par Marc et par le projet européen TIMECOP-AE conduit par Safran Turboméca, dont la tache "Operational evaluation of Eulerian and Lagrangian approaches" était dirigée par Marc et Julien Réveillon. Je remercie bien sûr les différents organismes qui ont permis que la thèse se passe dans de très bonnes conditions matérielles, mais surtout Marc et Frédérique qui ont dépensé un temps et une énergie remarquables pour monter les dossiers de financement et de projets. Je tiens à remercier particulièrement Marc qui, avec l'aide

du laboratoire EM2C et du département mathématiques du CNRS, a permis que la fin de thèse se passe dans des conditions idéales.

Une thèse c'est aussi un laboratoire d'accueil. Je remercie Nasser Darabiha de m'avoir accueilli au sein de laboratoire EM2C. Je remercie bien sûr tout le personnel administratif que j'ai croisé : Karine, Merryl, Véronique, Virginie, Anne-Cécile. En utilisateur intensif de l'informatique je remercie bien sûr Gilbert, ardent promoteur de la pomme. Un remerciement spécial va à Romain Pierrat, administrateur linux à ses heures perdues, pour tout le travail qu'il a fourni et qui a permis, entre autres, aux irréductibles linuxiens d'avancer dans de bonnes conditions. L'informatique au laboratoire EM2C lui doit beaucoup. Enfin je remercie toutes les personnes qui ont oeuvré pour que le laboratoire se dotte de moyens de calcul conséquents. Certaines réalisation de mes travaux de thèses n'auraient pu être menées à bien sans le cluster du laboratoire.

Il est très souhaitable, et ce fut ô combien mon expérience, que le travail de thèse ne soit pas synonyme d'isolement d'un étudiant, cloitré dans un bureau pendant trois années, mais bien d'un travail d'équipe. J'ai eu la chance d'évoluer, au cours de ces trois ans, dans une équipe, une équipe exigeante, qui pousse sans cesse à se dépasser et qui, en fixant des objectifs parfois au-delà du raisonnable, nous permet d'atteindre des réalisations au-delà de nos espérances.
Je remercie donc tout d'abord Sébastien Candel, à l'origine de la mise en place de cette équipe de mathématiciens appliqués au sein du laboratoire EM2C. Je remercie ensuite ses deux piliers, Frédérique et Marc, mes directeurs de thèse. Je retiens de leur encadrement une disponibilité constante, une exigence de rigueur infaillible, une excellence scientifique venant à bout de toutes les singularités et injectant sans cesse de nouvelles pistes d'investigation. De nombreuses qualités pourraient encore être listées, mais je voudrais insister sur trois points qui me paraissent essentiels. Tout d'abord la liberté qu'ils m'ont laissée pour orienter mes travaux de thèse représente pour moi une richesse majeure de ces trois années. De plus cette liberté était associée à une grande confiance, qui m'a permis d'avancer et de me construire pendant la thèse. Finalement, l'atmosphère de travail qu'ils créent au sein de l'équipe m'a permis de m'épanouir complètement pendant la thèse. Les nombreuses fois où la plus sérieuse des réunions se transforme subrepticement en un joyeux moment de camaraderie, autour de grands classiques du rock, du jazz ou autour de la chanson du dimanche, restent pour moi des moments forts.
Je remercie également Julien Réveillon, qui a co-encadré mes travaux. Mes visites à Rouen furent toujours très agréables, loin du stress de la région parisienne. Julien a été le catalyseur et l'inspiration de mes travaux de développements du code MUSES3D. De cette fructueuse collaboration est née le couplage entre MUSES3D et le solveur ASPHODELE, permettant les réalisations présentées dans la quatrième partie de ce manuscrit. Je salue au passage Bruno Delhom et Olivier Thomine qui ont contribué au développement d'ASPHODELE.
Ce travail d'équipe a été complètement démultiplié par l'arrivée de Lucie Fréret pour un stage post-doctoral au laboratoire EM2C. Nombreux sont les doctorants qui ont souhaités un jour avoir l'aide d'un stagiaire pour avancer ses travaux. En revanche, l'aide d'un docteur en calcul scientifique à temps plein sur les problématiques de la thèse, c'est un rêve que ne ferait même pas le plus dormeur des doctorants ! Nous avons mené, durant presque deux ans, une collaboration étonnante basée sur une complémentarité parfaite. La remercier semble bien peu tant elle a oeuvré dans cette entreprise. Elle a transformé l'essai calcul scientifique de la thèse. Afin de mesurer l'ampleur de sa contribution, il faut savoir que les chapitres 5, 7 et 11 de ce manuscrit n'auraient pu être réalisés sans son aide.

Encore une fois merci Lucie.
J'ai également eu la chance de pouvoir sortir un peu de notre univers virtuel et numérique pour collaborer, au sein même du laboratoire, avec une équipe d'expérimentateurs. Je remercie donc Corine Lacour, Daniel Durox et Sébastien Ducruix véritables chevaliers JEDI, ainsi que Patrick Duchaine, Thierry Schuller et Laurent Zimmer pour leurs doigts de FEE. Un coup de chapeau au passage à Marc, en quelque sorte le maître Yoda de ce joli projet. Enfin une dernière collaboration a marqué mes travaux de thèse : celle avec Prof. Rodney. O. Fox de l'Iowa State University. Je tiens à saluer sa gentillesse, sa parfaite maîtrise du français ainsi que son impressionante vitesse de croisière dans ses recherches, qui nous poussait à nous dépasser sans cesse pour ne pas rester trop loin derrière.
Finalement je souhaite bon courage aux actuels doctorants de l'équipe mathématiques appliquées du laboratoire : Damien Kah, Max Duarte et François Doisneau.

Une thèse se conclue par un manuscrit et une soutenance. Je remercie tout d'abord Bruno Després et Thierry Poinsot d'avoir rapporté sur mes travaux. Merci pour le temps qu'ils ont passé sur ce manuscrit, et je pèse mes mots car j'ai conscience qu'il n'est pas toujours facile à lire. Merci pour leurs questions, leurs conseils et l'intérêt qu'ils ont manifesté pour mes travaux. Rapporter sur des travaux pluridiscplinaires n'est sans doute pas chose aisée et avoir l'avis de spécialistes des deux disciplines au coeur de la thèse corrobore la démarche entreprise. Je les remercie tous les deux pour cela. Merci à Vincent Giovangigli d'avoir présidé mon jury de thèse. Aux antipodes d'un président pédant et grincheux que l'on pourrait imaginer dans une fiction, il a amené à la soutenance l'humour, le dynamisme et l'entrain qui le caractérisent. C'est bien sûr un honneur pour moi qu'il aie présidé le jury. Je remercie Sébastien Wey, correspondant DGA, ainsi que Thomas Lederlin, ingénieur Turboméca, pour l'intérêt qu'ils ont porté à mes travaux et pour leur présence à ma soutenance. Un grand merci à Rodney pour sa présence, il détient sans nul doute le record de la plus longue distance parcourue pour nous rejoindre. Enfin un grand merci à Jean-Michel Dupays pour sa présence, son intérêt pour mes travaux et pour ses nombreuses remarques et corrections issues d'une relecture minutieuse de la partie calcul scientifique (III) de ce manuscrit. Il a largement contribué à la valorisation de cette partie de mon travail de thèse.
J'en viens donc naturellement à remercier les courageux relecteurs du manuscrit : Marc, Frédérique, Julien, Lucie. Merci donc à Jean-Michel Dupays pour la relecture de la partie calcul scientifique, à Matthieu Boileau pour la relecture de la partie modélisation et à Daniel Durox pour la relecture du chapitre consacré aux comparaisons calcul/expérience. Merci à papa qui a relu l'introduction et un immense merci à mon épouse, Verónica, qui a passé l'intégralité du manuscrit au correcteur orthographique. Merci à Romain Pierrat et à Franck Richecoeur pour leur aide dans la phase d'édition de ce manuscrit.
Je souhaite également remercier Roland Masson, directeur du département Mathématiques Appliquées de l'IFP, qui m'a permis de consacrer du temps dans mes nouvelles activités pour achever les derniers travaux en cours, et en particulier la finalisation du manuscrit.
Un grand merci à ma famille pour l'organisation du pot. Je remercie particulièrement Véro et Belle-maman pour le magnifique assortiment de spécialités portugaises qu'elles nous ont concocté et qui fut très apprécié. Merci également à papa qui a su déplacer le débat Eulérien/Lagrangien vers des considérations oenologiques ! Merci à mes frères Luc et Eric pour leur présence qui m'a touché. Enfin merci à ceux qui ont permis que Sarah soit là et qui ont dû maintes fois traverser le laboratoire au pas de course pour essayer de la suivre afin d'éviter qu'elle allume un banc d'essai de combustion.

Cette thèse fut également faite d'échanges avec les membres du laboratoire EM2C. Je retiens mes discussions calculs avec Philippe Rivière, dont je salue l'impressionnante maîtrise de la norme FORTRAN. Je repense également aux échanges avec Nicolas Tran, en constante lutte pour l'amélioration des conditions de travail des thésards. Je salue bien sûr Romain Pierrat, mon guide dans la communauté des amis du pingouin, qui soutenait le dernier ilôt de résistance face à la domination de la pomme. Merci à Anne Bourdon, pour son attention et son écoute. Enfin je retiens les échanges avec Damien Kah, qui se poursuivent maintenant à l'IFP. Je ne peux, ayant déjà me semble-t-il largement dépassé la longueur conventionnellement admise pour des remerciements, énumérer la liste de tous les doctorants du laboratoire que j'ai eu plaisir à cotoyer, qu'ils soient tous remerciés.

Finalement, et j'achèverai par là ce long monologue, la thèse est indissociable de la vie du doctorant. Elle l'impacte fortement par ses exigences, et elle ne peut se faire sans tout ce qui nous donne la force d'avancer. Je tiens donc maintenant à remercier mes deux sources d'énergie dans la vie.
Je pense tout d'abord à mes amis qui représentent une source de joie intarissable. Merci à toute la bande de la BJ, félicitations aux nombreux jeunes mariés de 2008 et toute ma tendresse va aux petits trésors arrivés en 2009. Un merci tout particulier à Boni, venu d'Allemagne pour assister à la soutenance.
Mon dernier remerciement, et qui me tient tout particulièrement à coeur, est pour ma famille. Un premier merci général à tous pour l'engouement qu'a suscité ma thèse et que j'ai particulièrement ressenti autour de la soutenance. Merci à papa et maman qui ont su nous apprendre que l'essentiel dans la vie était l'accueil des plus petits, non seulement par la théorie, mais surtout par la pratique au jour le jour, avec l'accueil d'Olivier. Merci à mon petit frère Olivier pour son amour débordant. Félicitations à mes frères Luc et Eric pour leur récentes obtention de postes dans des domaines fondamentaux pour la société : la recherche et l'éducation spécialisée. Merci à mes beaux-parents sans qui je n'aurai pu rencontrer la personne la plus importante de ma vie. O acolhimento que eu recebi na sua familia, ca em França o la em Portugal foi realmente incrível. Enfin je remercie celle qui supporte mon caractère simplet et grincheux depuis dix ans, ma femme depuis cinq ans déjà, Verónica. Je lui dois les deux lumières qui éclairent ma vie. Tout d'abord sa compagnie de tous les instants, faite d'écoute, d'attention, de soutien mais aussi d'humour et de fous rires. Et puis bien sûr nos enfants, aujourd'hui Sarah et Nathanaël que j'embrasse tendrement.
Vous êtes le sens de ma vie.

Un personnage manque à l'appel dans ces remerciements, ses six collègues sont mentionnés mais malgré mes efforts ce dernier nom, en forme d'onomatopé, n'a pu trouver sa place dans ces lignes, qu'il m'en excuse. La signification de cette obscure allégation est à chercher du côté d'un des plus universel contes des frères Grimm.

Voilà, je remercie maintenant par avance le lecteur, à qui je souhaite bon courage. Je crois que la longueur de ces remerciements lui donne une idée de ce qui l'attend dans la suite du manuscrit...Qu'il se rassure néanmoins, l'anglais permet une formulation plus synthétique ! Qu'il ne désespère donc pas et qu'il se rappelle ce qu'Edmond Rostand disait :

C'est la nuit qu'il est beau de croire à la lumière.

Résumé

Le modèle multi-fluide permet de décrire par une approche Eulérienne les sprays polydispersés et apparaît donc comme une méthode indiquée pour les applications de combustion diphasique. Sa pertinence pour la simulation à l'échelle d'applications industrielles est évaluée dans ce travail, par sa mise en oeuvre dans des configurations bi-dimensionnelle et tri-dimensionnelle plus représentatives de ce type de simulations. Cette évaluation couple une étude de faisabilité en terme de coût de calcul avec une analyse de la précision obtenue, par des comparaisons avec les résultats de méthodes de références pour la description des sprays. Afin de définir une telle référence, une hiérarchisation des modèles de spray est proposée dans ce travail, soulignant les niveaux de modélisation associée aux diverses méthodes. Une première configuration d'écoulements tourbillonnaires est utilisée pour caractériser la méthode multi-fluide. L'étude de la structure mathématique du système de lois de conservation permet d'analyser la formation de singularités et de fournir les outils permettant d'évaluer leur impact sur la modélisation. Cette étude permet également de dériver un schéma numérique robuste et efficace pour des configurations bi- et tri-dimensionnelle. La description des dynamiques de gouttes conditionnées par la taille est évaluée dans ces configurations tourbillonnaires au moyen de comparaisons quantitatives, sur des champs instantanés, où le multi-fluide est confronté à une méthode Lagrangienne, ainsi qu'à des résultats expérimentaux. Afin d'évaluer le comportement de la méthode multi-fluide dans des configurations plus représentatives des problématiques industrielles, le solveur MUSES3D est développé, permettant, entre autres, une évaluation fine des méthodes de résolution des sprays. Une implémentation originale de la méthode multi-fluide, conciliant généricité et efficacité pour le calcul parallèle, est réalisée. Le couplage de ce solveur avec le code ASPHODELE, développé au CORIA, permet d'effectuer une évaluation opérationnelle des approches Euler/Lagrange et Euler/Euler pour la description des écoulements diphasiques à inclusions dispersées. Finalement, le comportement de la méthode multi-fluide dans des jets bi-dimensionnels et dans une turbulence homogène isotrope tri-dimensionnelle permet de montrer sa précision pour la description de la dynamique de sprays évaporant dans des configurations plus complexes. La résolution de la polydispersion du spray permet de décrire précisément la fraction massique de combustible en phase vapeur, un élément clé pour les applications de combustion. De plus, l'efficacité du calcul parallèle par décomposition de domaine avec la méthode multi-fluide permet d'envisager son utilisation à l'échelle d'applications industrielles.

Mots-clés Écoulements diphasiques; Sprays polydispersés; Méthode multi-fluide; Systèmes de lois de conservation faiblement hyperboliques; Schémas numériques cinétiques; Informatique scientifique; Calcul parallèle.

Summary

The multi-fluid model, providing a Eulerian description of polydisperse sprays, appears as an interesting method for two-phase combustion applications. Its relevance as a numerical tool for industrial device simulations is evaluated in this work. This evaluation assesses the feasibility of multi-fluid simulations in terms of computational cost and analyzes their precision through comparisons with reference methods for spray resolution. In order to define such a reference, the link between the available methods for spray resolution is provided, highlighting their corresponding level of modeling. A first framework of 2-D vortical flows is used to assess the mathematical structure of the multi-fluid model governing system of equations. The link between the mathematical peculiarities and the physical modeling is provided, and a robust numerical scheme efficient for 2-D/3-D configurations is designed. This framework is also used to evaluate the multi-fluid description of evaporating spray size-conditioned dynamics through quantitative, time-resolved, comparisons with a Lagrangian reference and with experimental data. In order to assess the multi-fluid efficiency in configurations more representative of industrial devices, a numerical solver is designed, providing a framework devoted to spray method evaluation. An original implementation of the multi-fluid method, combining genericity and efficiency in a parallel framework, is achieved. The coupling with a Eulerian/Lagrangian solver for dispersed two-phase flows, developed at CORIA, is conducted. It allows a precise evaluation of Euler/Lagrange versus Euler/Euler approaches, in terms of precision and computational cost. Finally, the behavior of the multi-fluid model is assessed in 2D-jets and 3-D Homogeneous Isotropic Turbulence. It illustrates the ability of the method to capture evaporating spray dynamics in more complex configurations. The method is shown to describe accurately the fuel vapor mass fraction, a key issue for combustion applications. Furthermore, the method is shown to be efficient in domain decomposition parallel computing framework, a key issue for simulations at the scale of industrial devices.

Keywords Two-phase flows; Polydisperse sprays; Multi-fluid method; Weakly hyperbolic systems of conservation laws; Kinetic numerical schemes; Scientific computing; Parallel computing.

Contents

List of Figures

Introduction

Industrial combustion device efficiency has continuously increased for the past decades. This progress was, for its main part, due to major advances on the device technology. In the aeronautical sphere for example, the advent of double-flow turbo engine, where an important part of the coming flow in the engine is not burnt, led to a great reduction of fuel consumption. In the automotive field, the reduction of the vehicle weight had an important impact on fuel consumption reduction. Besides the development of auto-ignition engine, due to the advent of turbo-compressor overfeeding and to common rail injection, gives an interesting alternative for a reduction of engine consumption, nevertheless associated with a higher pollution level. Consumption reduction, as well as pollutant emission reduction, in combustion devices, are to be continued and represent a crucial issue since combustion still produces 88% of the world primary energy. The challenge is now to improve the combustion process itself, developing for example lean combustion, i.e., with a fuel deficiency, in order to reduce significantly its consumption. Nevertheless, these techniques are favourable to the development of combustion instabilities, (Poinsot and Veynante 2005), that can lead to extinction, to a higher pollutant production (Schmitt, Poinsot, Schuermans, and Geigle 2007), or even to the damaging or destruction of the device. A very precise understanding of the full combustion process is thus required to be able to foretell such instabilities outbreak.

The physical processes involved in combustion are complex, and their coupling with a turbulent flow, occuring in many industrial devices, makes it more difficult. Indeed the description of the flow turbulence is required as well as the description of the mixing, the ignition and the combustion itself, i.e., a complex system of chemical reactions, in a context where, often, the type of combustion regime, premixed, partially premixed or diffusion (Poinsot and Veynante 2005), is not known. Furthermore, the interaction between flow and combustion can also be coupled with the acoustics of the system, (Candel 2002; Noiray, Durox, Schuller, and Candel 2008), leading to higher complexity. To have an access to this physics, theoretical studies are carried through, as well as experiments and numerical computations. Given the cost and the complexity of experiments in a combustion context, there is a great impetus in developing numerical computations. Besides, the simulation gives access to the complete description of the problem. Nevertheless, the advent of predictive simulations cannot be separated from the development of experimental studies, required for the simulation tool validation. As defined by (Pope 2000), two kinds of approaches can be distinguished for numerical simulation of turbulent flows:

- turbulence modeling approaches, as Reynolds Average Navier Stokes (RANS) methods or Probability Density Function (PDF) methods,

- turbulence simulation approaches, as Direct Numerical Simulation (DNS) or Large Eddy Simulation (LES).

The same classification applies to turbulent combustion simulation (Poinsot and Veynante 2005). In industry, turbulent model approaches are mainly used, given their lower cost compared to simulation approaches. They give a satisfactory answer for steady configurations. Nevertheless, due to the high unsteadiness of instability phenomena, it appears necessary to develop a simulation-based approach. While DNS, where all the scales of the problem are resolved, cannot, at present, be used in an industrial configuration given its cost, LES simulation has been recently shown to be efficient for combustion at industrial scale to reproduce, for example cyclic variations of spark-iginition engine (Richard 2005), combustion instabilities (Rey 2004; Selle, Benoit, Poinsot, Nicoud, and Krebs 2006; Martin, Benoit, Sommerer, Nicoud, and Poinsot 2006) or iginition (Boileau 2007). A high maturity, associated to very efficient computations, has been reached in the field of monophasic gaseous combustion. Nevertheless, two-phase combustion cases still lead to major difficulties. Efforts must then be devoted to two-phase combustion, since it is very often encountered in industrial devices. This preoccupation is the guideline of the European Community project: Toward Innovative Method for Combustion Prediction in Aero-Engines (TIMECOP-AE). This project aims at significantly improving the state of art of two-phase combustion prediction methods.

Indeed, in many industrial applications, the fuel is stored in liquid form, since this condensed form is easier to store and safer. In auto-ignited, or direct injection spark-ingnited, automotive engines, aeronautical turboshaft or turbojet engines, a two-phase flow combustion takes place following a liquid fuel injection. Two phase effects significantly influence flame structures and pollutant production. Therefore a great attention must be devoted to the presence of the liquid phase.
Two-phase interactions as well as liquid influence on the combustion process, are the main spray effects that must be accounted for.

- As far as **phase interactions** are concerned, coupled transfers occur between gas and liquid phases in the course of the fuel injection.
  - The first phenomenon encountered is the break-up of the liquid core injected. This phenomenon, called **primary break-up**, leads to a discontinuous liquid phase constituted of ligaments, clusters and droplets with a large size range. During this stage, liquid and gas phases exchange mainly momentum. Indeed, the gas phase can contribute to the break-up of the liquid core, as in air-assisted atomizers (Lefebre 1989). With regard to the liquid phase, air entrainment by the jet can occur, particularly in pressure atomizer where the liquid is injected with a high velocity (Lefebre 1989).
  - Afterward, the gas phase interacts with the resulting discontinuous liquid phase. Momentum transfers still occur through interactions between the spray and the gas turbulent eddies. These interactions may lead to the secondary break-up of the liquid ligaments and clusters into smaller spherical droplets. These momentum transfers may come with heat transfers. The liquid phase is heated and then vaporizes, leading to a mass transfer from the liquid to the gas.

  Moreover, liquid-liquid interactions may occur, as collision or coalescence, acting upon the physics of the liquid phase.

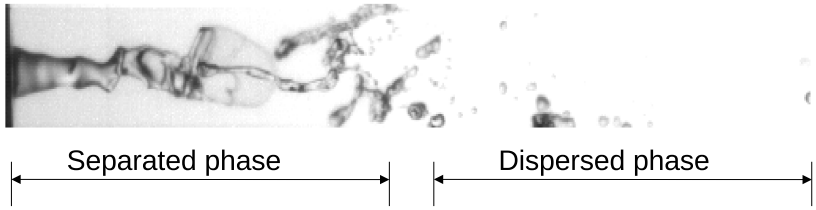


Figure 1 – *Break-up steps for a liquid injection: from a liquid core to a droplet spray. Source: Y. Merry, EM2C Laboratory.*

- The **combustion** of a liquid spray is significantly different from the purely gaseous one. If we consider that the spray is completely vaporized before burning, the combustion is homogeneous. This situation is often taken as a simplification assumption and can occur if the vaporizing and burning zones are separated. Otherwise, if combustion and vaporization are mixed, different spray combustion regimes are defined. A first distinction is made in (Borghi and Destriau 1995), between external combustion, where the flame is located around the spray, and internal combustion, where each droplet burns separately. The group combustion regime is introduced in (Candel, Lacas, Darabiha, and Rolon 1999). In (Reveillon and Vervisch 2005), the influence of equivalent ratio and advection of the spray on combustion is taken into account. Different types of external combustion are characterized. Besides, new regimes are introduced, as group combustion of droplet clusters and hybrid regime, combining external and group combustion. Furthermore, two-phase combustion depends on the fuel vapor repartition, driven by liquid evaporation. The dynamics of the liquid phase thus directly influences combustion, since it influences vapor fuel repartition.

To obtain information on the physics of two-phase flow combustion, as done for purely gaseous combustion, theoretical, experimental and numerical works are conducted. Nevertheless, the presence of the liquid phase introduces further complexity in experiments, and thus the need of numerical simulations is even greater.
The key question appearing at present is the simulation of a full spray injection. This question cannot be treated easily as two different zones clearly appear for the injected liquid phase, needing different types of modeling.

- In the primary break-up zone, Fig. 1, the liquid phase being continuous, it is necessary to adopt a **separated-phase** flow description. In this framework, the shape of the liquid phase in the gas flow is not presumed. The liquid can thus be continuous or discontinuous with ligaments and droplets. This description is more naturally associated with a Eulerian description of the liquid phase, where the evolution of mean quantities, i.e. density, velocity and energy of the liquid phase, are computed at fixed points. The separated-phase formulation has been first formalized in (Delhaye and Achard 1977; Ishii 1975; Marle 1982). This approach is used for thermo-hydraulic modeling, (Allaire, Clerc, and Kokh 2002), and for spray injection in combustion applications, (Iyer 2001; Truchot 2005)

- On the other hand, in the secondary break-up zone, Fig. 1, a **dispersed-phase** flow description is more appropriate. In this case, the liquid phase is assumed to be composed of spherical droplets. Furthermore, the droplets are considered as point particles, so that the gas flow is not resolved around and inside droplets, and the coupling with the gas phase is done through source terms in the gaseous phase equations.

The liquid phase is thus assumed to be dilute, that corresponds, in the classification established in (O'Rourke 1981), to a liquid volume fraction α_l, going from $\alpha_l = 10^{-4}$, thick spray, to $\alpha_l = 10^{-2}$, dilute or thin spray. Moreover, droplet size must be smaller than the smallest scale of the gas turbulence, the Kolmogorov length scale, (Kuo 1986). In this framework, a Lagrangian tracking approach is often used, where the information, i.e., droplet characteristics, is tracked in the flow.

Therefore, important efforts have been devoted to modeling and simulation of these two different zones arising in a liquid injection. The key issue in describing primary break-up is to obtain the correct size distribution created as well as the size-velocity correlations. On the other hand, the simulation of the dispersed phase must predict the time-resolved dynamics of a polydispersed droplet liquid phase interacting with the gas phase. A crucial challenge at present is to couple these descriptions in order to carry out a full injection computation.

In industrial framework, full liquid injection computations are done through two methods:

- a dispersed-phase Lagrangian approach is used, assuming that the liquid phase is constituted of droplets, even in the region close to the injector nozzle. In this case, primary break-up is not described and models are required. It is thus difficult to obtain the correct size distribution. This liquid description is coupled with a RANS description of the gas flow, (Amsden, O'Rourke, and Butler 1989).
- On the other hand, recent developments have led to computation of a full liquid injection using Eulerian separated-phase description. In this case, a RANS resolution of the gas and of the liquid is done as well as the resolution of an equation for the liquid volume fraction. This approach can be found for example in the industrial code C3D, developed at IFP, (Truchot 2005; Vessiller 2007), or in (Iyer 2001). In this RANS computation, the polydisperse character of the spray cannot be accounted for, the liquid system of equations being resolved for only one typical size of droplets. This model is of two-fluid type.

An interesting answer to the limitation of these two types of simulation has been provided with the coupling of an Eulerian separated-phase description for the simulation of the liquid core primary break-up with a Lagrangian description of the dispersed phase, in regions distant from the injector nozzle, done in the solver Eulerian Lagrangian Spray Atomization (ELSA), (Demoulin, Blokkeel, Mura, Beau, and Borghi 2007).

Such a full spray injection computation in a LES framework is today an essential contribution toward a predictive two-phase combustion simulation. Two main research directions are followed at present with this objective.

- First, precise DNS of the primary break-up are conducted in order to extract more precise information than average values of the resulting droplet size and velocity distributions. In this case, two systems are resolved by a DNS, one for the gas and one for the liquid, and the interface between the liquid and the gas phase is completely resolved. It can be done with a Eulerian method, using Volume Of Fluids (VOF) and Level Set methods, (Menard, Tanguy, and Berlemont 2007). The liquid jet primary break-up plotted in Fig. 2, is computed with this Eulerian DNS method.

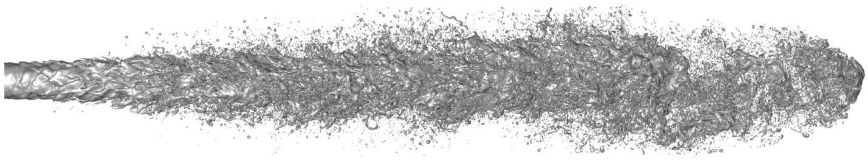


Figure 2 – *Liquid jet primary break-up computed with a Eulerian description combining VOF and level set method to completely resolve the interface between the liquid and the gas phase, (Menard, Tanguy, and Berlemont 2007).*

The evolution of the interface can also be done with a Lagrangian tracking, (Popinet and Zaleski 1999).

- Second, studies are conducted in order to choose the more efficient resolution method for the LES of the polydispersed liquid phase. These studies concern for example Eulerian methods, that are of great interest for massively parallel computations.

The coupling of these two types of models in a LES context is still an open question, for example concerning the type of information that should be unherited from the primary break-up simulation in the dispersed-phase simulation.

The present work focus on resolution methods for the dispersed liquid phase. Several strategies, of heterogeneous types, are available for the dispersed liquid phase modeling. It is often difficult to make the link between these approaches, and we provide here a short classification.
Two level of modeling can be envision for the dispersed-phase, based on a deterministic or on a probabilistic approach.

- A deterministic approach can be conducted for the simulation of the liquid phase. In this case all the droplets of the spray are tracked in the flow, using a Lagrangian method. We refer to this method as the Discrete Particle Simulation (DPS), (Mashayek 1998; Reveillon and Vervisch 2005; Miller and Bellan 2000). Numerical particles representing one droplet are tracked in the flow, through the resolution of their characteristic evolution: size, center of mass position, velocity and temperature. This method can be coupled with different descriptions of the gaseous flow: DNS, LES or RANS. In the case of the coupling with a gas DNS, it is often considered as a two-phase flow DNS. Nevertheless one has to keep in mind the modeling assumptions associated with the dispersed-phase description. When coupling with gas LES or RANS simulations, further modeling is to be introduced to take into account modeled gas scales, and turbulent dispersion terms are introduced.

- On the other hand, a probabilistic formulation can be introduced to model liquid dispersed-phase. In this case a number density function (NDF) of the spray f^ϕ is introduced, the quantity $f^\phi(t,x,S,u,T)\mathrm{d}t\,\mathrm{d}x\,\mathrm{d}S\,\mathrm{d}u\,\mathrm{d}T$ being the probable number of droplets with, at time t, a position in $[x, x+\mathrm{d}x]$, a surface in $[S, S+\mathrm{d}S]$, a velocity in $[u, u+\mathrm{d}u]$ and a temperature in $[T, T+\mathrm{d}T]$. The NDF satisfies a Williams-Boltzmann

equation:

$$\underbrace{\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{u} f)}_{\text{Free Transport}} + \underbrace{\partial_{\boldsymbol{u}} \cdot (\boldsymbol{F}\, f)}_{\text{Drag Force}} - \underbrace{\partial_S (R_s\, f)}_{\text{Evaporation}} + \underbrace{\partial_T (E f)}_{\text{Heat Transfer}} = \underbrace{\Gamma}_{\text{Collisions}} + \underbrace{Q}_{\substack{\text{Secondary} \\ \text{Breakup}}} . \tag{1}$$

In this framework, along with the dispersed-phase assumptions, we further assume that the effect of the gas on the liquid only depends on local gas properties, i.e., there is no long-distance interactions. This approach is called the kinetic model, by analogy to kinetic theory of gases. It is a Eulerian statistical approach. Nevertheless, given the high dimension number of the phase space (8 in 3-D) a finite volume discretization of this equation, called the "full spray equation method" in (O'Rourke 1981), cannot be used at the scale of an industrial burner. Resolution methods have thus to be derived from this kinetic model, to obtain macroscopic models, well-suited for numerical simulation at industrial scale. Two resolution strategies are used:

- a stochastic Lagrangian Monte-Carlo method called Direct Simulation Monte Carlo method, (Bird 1994; Hylkema and Villedieu 1998; Hylkema 1999), where stochastic parcels are tracked within the flow. The link between Lagrangian statistical methods and DPS has to be clearly highlighted. Indeed, statistical Lagrangian description can refer to two different levels of modeling:

  1. the most common one is associated with a coarser version of DPS, needed in industrial configurations for computational cost reasons. In this framework, a numerical particle, or parcel, represents several physical droplets, (Dukowicz 1980; O'Rourke 1981). In this case, the computed solution does not approximate, or approximate with a lot of noise, the spray NDF and its moments, defined at the kinetic level of description. This method is refered in this work as the Stochastic Parcel (SP) method, (O'Rourke 1981).

  2. the second one, that we call DSMC in this work, is a resolution method for the kinetic equation Eq. (1), as introduced in (Bird 1994) for rarefied gas. In this framework, we aim at achieving a converged solution approximating the NDF moment dynamics. It is thus equivalent to the Eulerian methods derived from the kinetic model. It represents a refined vision of the DPS, several statistical particles being needed for one physical droplet. The numerical particle has a weight associated, adapted to the needed refinement.

  Practically speaking, the difference between DPS and statistical methods arises in collision modeling. The stochastic particles allow to compute directly the collisional integrals in a statistical way, whereas collisions are seen deterministically in DPS. Besides the different visions of the statistical methods described above leads to a major difference in the refinements used, since a converged solution, in the sense of the kinetic modeling, is expected in DSMC method.

- A wide range of Eulerian techniques are derived from the kinetic model. Various techniques are used to derive a system of equations governing the evolution of a set of moments of the spray NDF. The most common methods, for spray simulation, are two-fluid models (for dispersed phase), (Fevrier 2000; Kaufmann

2004), multi-fluid models, (Greenberg, Silverman, and Tambour 1993; Laurent and Massot 2001), and the Direct Quadrature Method of Moments (DQMOM), (Marchisio, Vigil, and Fox 2003).

The kinetic model, and resulting methods, can be used for different gas descriptions. Nevertheless, if LES or RANS descriptions are used, a kinetic model "in the mean" has to be derived to take into account unresolved scales, leading to a new unclosed Williams equation, different from Eq. (1), (Reeks 1991). Further modeling is thus needed to close the equation.

At present, Lagrangian methods are widely used for the dispersed-phase simulation since they combine an efficient modeling of the polydisperse phase, a high numerical efficiency, not introducing any numerical diffusion, and an easiness of implementation. Nevertheless, they introduce a difficult coupling with the Eulerian described gas phase, and if a DSMC is considered, a very high number of statistical particles has to be used for unsteady polydisperse cases. Furthermore, in the framework of domain decomposition for parallel computations, it is needed to use complex and costly dynamic partitioning methods, to ensure a good load balancing between the different parallel processes. Recent advances have been obtained in this field (Garcia 2009). Eulerian methods for dispersed-phase modeling, derived from the kinetic model, provide thus an interesting alternative to Lagrangian methods, since they can easily be coupled with the gas phase and are well-suited for massively parallel computations. Nevertheless, the key issue of size polydispersion description represents a great difficulty in the framework of Eulerian methods. This issue has been adressed in the field of Chemical Engineering, for aerosol dynamics and aggregation-breakage processes. In this field the dispersed phase is modeled by a Population Balance equation (Ramkrishna and Fredrickson 2000), equivalent to the spray Williams equation. Efficicient methods have been designed to describe polydispersion for mono-variate cases, with only one parameter for particle size, and no velocity distribution, (McGraw 1997; Ramkrishna and Fredrickson 2000; Marchisio, Vigil, and Fox 2003). Extension to multi-variate distributions have been recently conducted in (Marchisio and Fox 2005; Fox 2008). At present, solely the multi-fluid model and the DQMOM provide a satisfactory answer for size distribution description in a spray framework, where a multi-variate distribution function is required. Furthermore, a second drawback of Eulerian method is the difficulty to describe spray trajectory crossings. This problematic is related to the Knudsen number (Kn) of the considered flow, Kn being the ratio of the mean free path of droplets, $l_0 = 1/(\sigma_0 n_0)$, and a characteristic length, L_0 of the flow. This term arises in the dimensionless collision term, $\overline{\Gamma}$ formulation:

$$\overline{\Gamma} = \frac{\Gamma}{\mathrm{Kn}}, \quad \mathrm{Kn} = \frac{1}{\sigma_0 n_0 L_0}, \tag{2}$$

where σ_0 is the collision cross section and n_0 a typical droplet number density. Therefore, in an infinite Knudsen limit, there is no collision and spray clusters may thus cross each other, leading to droplet trajectory crossings occurence. Besides, for finite Knudsen number, collisions will occur but droplet clusters may only partially collide, still leading to droplet trajectory crossings. Although naturally described with Lagrangian methods, these trajectory crossings are very difficult, as explained in details in this work, to describe with Eulerian methods. Indeed, most of the Eulerian methods for spray provide a description in a zero Knudsen limit where the droplet velocities relax toward an equilibrium Maxwellian distribution, that is the hydrodynamic limit (Bardos, Golse, and Levermore

1991). Nevertheless, recent developments of Quadrature Method of Moments (QMOM), (Fox 2008), provide Eulerian description of droplet trajectory crossings, in the framework of solid particle two-phase flows. These techniques have been recently introduced in the multi-fluid method, (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009; Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009). The multi-fluid method is thus, at present, the only Eulerian method describing the spray polydispersion as well as droplet trajectory crossings. Its important drawback appears to be the cost that can arise from the size conditioning of the method, leading to the resolution of several systems of conservation laws, one for each of the size discretization interval used. Nevertheless, it appears interesting, in the actual context of massively parallel simulation, to devote interest to this method.

The multi-fluid method, first introduced in (Greenberg, Silverman, and Tambour 1993), is based on a finite volume discretization of the size phase space. For each size interval, called section, a system of conservation laws is obtained through moment methods derived from the kinetic level of description. The precise derivation of the model, as well as the precise derivation assumptions, have been given in (Laurent and Massot 2001). Up to now, the multi-fluid has been shown to be efficient for:

- laminar polydispersed spray flames, (Laurent 2002b), through numerical, (Laurent and Massot 2001), or experimental, (Laurent, Santoro, Noskov, Gomez, Smooke, and Massot 2004), comparisons,
- dense sprays, (Laurent 2002b; Laurent, Massot, and Villedieu 2004; Fox, Laurent, and Massot 2008), with description of collisions leading to coalescence, and of secondary break-up, (Dufour 2005).

Furthermore, several mathematical studies focused on the multi-fluid method have been conducted about:

- the propagation of plane polydispersed spray flames, (Laurent 2002b);
- the mathematical structure of secondary break-up operator, (Dufour 2005);
- the link between the fluid and the kinetic level of description, (Dufour 2005; Massot 2007).

Finally, in order to improve the efficiency of the method for the size phase space description, an important attention has been devoted to the numerical method for the size variable:

- the method is shown to be at most of first order in the size variable, (Laurent 2002a);
- second order methods in the size variable are developed, (Dufour and Villedieu 2005; Laurent 2006);
- new size moment methods have been designed to obtain a high order resolution of the evaporation, (Massot, Laurent, Kah, and de Chaisemartin 2009).

All these studies have conducted to a Eulerian method providing size distribution, as well as size-velocity correlations, description. Moreover, it presents interesting modeling properties as collisions, coalescence and secondary break-up descriptions, with fine characterizations from a mathematical, a numerical and an experimental point of view. Nevertheless, these studies, mainly based on 1-D computations, for 1-D or 2-D configurations, never assessed the feasibility of multi-fluid computations in realistic configurations. We thus evaluate in

this work the relevance of the multi-fluid method as a numerical tool for industrial spray combustion computations.

To examinate a such question, two steps are considered in this work:

- characterize precisely the multi-fluid model in intermediate DNS configurations with well-controled flows, as 2-D vortical flows, achieving detailed comparisons with a reference Lagrangian resolution of the spray, and with experiments.
- Provide more complex, albeit still academic, configurations as DNS of 2-D or 2-D axisymmetric jets and of 3-D Homogeneous Isotropic Turbulence (HIT), to assess two crucial issues:
  - the feasibility of such simulations in terms of computational cost;
  - the ability to globally capture size-conditioned dynamics of the spray dispersed by a turbulent gas flow.

One can see that we do not tackle the issue of computation at an industrial scale with LES, since we stay with academical DNS configurations. Nevertheless, this is a needed first step to ensure that multi-fluid has the required robustness and precision as well as a reasonable computational cost, to be used afterward in an industrial scale context. Furthermore, this DNS context provides the basis needed to derive turbulent modeling for LES description.
To conduct such studies, new numerical developments are to be done. Indeed we need to:

- provide a precise and robust numerical method for 2-D, 2-D axisymmetric and 3-D configurations, with the best compromise between stability, precision and computational cost;
- provide efficient numerical tools for multi-fluid computations, combining a high genericity and efficiency in parallel distributed memory framework.

The need to develop numerical tools can seem questionable, given the number of highly efficient numerical platforms existing today. Nevertheless the need of academic simulation platforms still remains. Indeed, the evaluation of new methods in academic, albeit reasonably realistic, configurations, is a needed first step before integration in semi-industrial or industrial solvers. Furthermore, taking advantage of the relative simplicity of these configurations allows to achieve this objective very fastly and efficiently. Yet, the important evolution of scientific computing requires an important evolution of the academic research numerical tools, that must offer a high genericity and be efficient in parallel architectures.

An important feature we are aiming at in the present study, is genericity. Indeed, we want to study the applicability of multi-fluid method for industrial applications, so even if we need to define a simpler context for the purpose of the study, the method, as well as its numerical tools, must not reduce to this context. Three main types of extensions could be considered from the method used in the present study when addressing industrial devices simulation, with LES.
The first point concerns modeling. In a LES context, some scales of the gas flow are modeled, therefore we discuss in this work the extension of the kinetic basis to such gaseous flows. Furthermore, in this context, more complex droplet models, for example with advective corrections and droplet saturation description will be required. A complete modeling framework is thus presented in this work, even if using some simplifications in the cases studied.

Second, the numerical schemes provided have to be extendable to such configurations. One has to note that the numerical schemes given in this study could be used in LES configurations, given their robustness and reliability. Their only restriction is to be limited, at present, to second order precision. Moreover, the numerical schemes provided enable complex droplet modeling.
Third, more complex configurations may require to use recent extensions of the multi-fluid method, for size or velocity phase space description, and thus the numerical tools have to be easily extendable to these outbreaking methods.

We therefore base this study on a Eulerian multi-fluid method, associated to a given modeling framework, as described in Part. I. The modeling framework is needed for multi-fluid evaluation purpose, but one has to keep in mind that the chosen method does not restrict to this framework. We focus in Part. II on the characterization of the chosen method, from a mathematical, a numerical and an experimental point of view. We tackle in Part. III, the scientific computing issues, needed to develop a numerical tool for efficient and generic parallel multi-fluid computations. Finally, we assess in Part. IV, the cost and the precision of the multi-fluid in more complex configurations, up to a droplet-laden 3-D HIT simulation.

The kinetic modeling for spray, and its coupling to the gas phase description are detailed in **chapter one**. Governing equations are given for both the dispersed-liquid and the gas phase in a detailed modeling framework. In order to conduct precise evaluation of the multi-fluid method, we define a simplified modeling framework for both the gas, where we assume low Mach or incompressible flows, and the liquid phase, where simple droplet models are used; one-way coupling is further assumed. We insist again on the fact that this framework is not a restriction. Furthermore, we define a procedure to assess, still in a context favourable to method evaluation, the efficiency of the liquid description for combustion applications. This kinetic level of modeling, called "full spray equation method" in (O'Rourke 1981), cannot be directly solved given the high number of phase space dimensions, 8 in 3-D. We thus present in **chapter two** the resolution methods derived from the kinetic model, and practicable at the scale of industrial devices. In this chapter, we highlight the link between the kinetic model and the different macroscopic Lagrangian and Eulerian methods. As far as Eulerian methods are concerned, we provide a classification based on the derivation techniques of the main methods available at present. We analyze the resulting modeling properties of these Eulerian methods. This analysis is necessary in order to highlight the link between the numerous recently outbroken Eulerian methods, (Marchisio, Vigil, and Fox 2003; Fox 2008; de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009). Finally, we detail the type of multi-fluid method used throughout this work. We justify the choice made of a robust, mature and relatively simple "classical", in a sense defined in this chapter, multi-fluid method by the will to move toward industrial applications. Again, this choice is not a restriction, and new multi-fluid developments could be included later on.

To conduct the complete characterization of the multi-fluid method in vortical flows, we first focus, in **chapter three**, on the mathematical structure of the multi-fluid governing system of equations. This step is a key issue as the system of equations can lead to mathematical singularities. The structure of such singularities as well as the consequence on the modeling have to be analyzed, to understand the multi-fluid behavior. One has to note that these singularities, also called $\delta-$shocks, are not linked only to the multi-fluid method, but to all the Eulerian methods based on presumed velocity distribution assumption for the spray. The study we conduct relies on the mathematical study of the link between kinetic and

fluid descriptions done in (Dufour 2005), for which we provide new illustrations based on more complex vortical flows. Moreover, we design in this chapter a second order numerical method, able to compute these singularities and well suited for 2-D and 3-D applications. The robustness of the designed scheme makes it a good candidate for a wide range of applications. Furthermore, efforts were made so that a broad range of droplet models can be used. This numerical scheme, as well as the multi-fluid method precision, are assessed in **chapter four** for vortical flows, through comparisons with DSMC Lagrangian computations, taken as a reference. The ability of the scheme to compute $\delta-$shock singularities and the physical impact of such singularities are provided. The precision of the multi-fluid method for describing size-conditioned dynamics is assessed through convergence toward Lagrangian reference study. Finally, the multi-fluid is compared, in **chapter five**, to experiments in a well-controled laminar vortical jet. This chapter provides size-conditioned comparisons between a polydisperse spray Eulerian description and experimental measurements, a novel element, as much from the numerical point of view, given the difficulty to describe polydispersion with Eulerian methods, as from the experimental point of view where innovative diagnostics, as Interferometry Particle Imaging (IPI), allow to reconstruct a Eulerian field conditioned by size.

These first characterizations allow us to move to more complex flows. In order to achieve computations for these cases, a new academic solver for the multi-fluid has been provided. The difficulty of coupling a high genericity level with a high computational efficiency was tackled in this development, since a generic numerical tool working efficiently in a parallel context is expected. The implementation and optimization of the multi-fluid model lead to original scientific computing issues. Indeed, the method is conditioned by droplet size and therefore a 3-D computational domain leads to a 4-D computation. We present in **chapter six** a way to implement generically, emulating object orientation programming, the multi-fluid method. This genericity is needed, in a research environment, in order to obtain a fast evolution, a concentration of the development efforts in the field, and an easy diffusion of the numerical tools developed. For the multi-fluid case, the ability to couple the solver with a gas solver, is an important issue. The main choices made for the development of the MUlti-fluid Solver for Eulerian Spray (MUSES3D) are explained in this chapter. They form an original approach of solver implementation, using the possibilities of Fortran90/95. To provide to this solver the required computational efficiency for 3-D applications, we complete, in **chapter seven**, its optimization. Two directions are studied:

- sequential optimization, based on memory optimization and cache access;
- parallel optimization, through domain decomposition.

This chapter provides precise studies on the domain decomposition strategies that can be chosen in the multi-fluid 4-D environment. Test cases as well as solver scalability and efficiency are assessed through computations done on the EM2C laboratory cluster MADNESS.

The computational code MUSES3D is used in more complex configurations to assess the cost and the precision of the multi-fluid method to compute the dynamics of a polydisperse spray. A first step toward combustion is also made. To conduct such computations, the multi-fluid solver must be coupled with a gas solver. The coupling completed with the ASPHODELE solver of the CORIA laboratory, described in **chapter height** leads to an original numerical tool including Eulerian/Eulerian and Eulerian/Lagrangian descriptions of dispersed two-

phase flow. This academic solver is particularly well suited for spray method evaluation as both Eulerian and Lagrangian descriptions can be used, in a one-way coupling framework, with the same gas computation, i.e., within the same code run. In **chapter nine**, we investigate the ability of the multi-fluid model to simulate the dynamics of a polydisperse evaporating spray injected in a 2-D axisymmetric jet. This configuration is used since it is closer to spray injection industrial configurations of interest. Qualitative comparisons with Lagrangian DPS evaluate the description of size-conditioned dynamics for evaporating and non-evaporating sprays. The DPS is chosen for the Lagrangian method, though it does not provide the same level of description as the multi-fluid method, in order to be closer to industrial applications. A key question that must be tackled to investigate the relevance of multi-fluid description for industrial configurations, is the associated computational cost. We highlight in this chapter the difficulty to compare the cost of two heterogeneous methods, as Eulerian and Lagrangian spray description. We insist on the fact that the considered methods must be related to their corresponding level of description before drawing any conclusions concerning their relative cost. Indeed, the compared methods should correspond to the same level of description, defined in the first part of the manuscript. We decide to present cost comparisons with two approaches:

- first we compare the cost of the multi-fluid to the one of DPS. Although these methods do not correspond to the same level of description, it appears necessary to compare the cost of the multi-fluid method to a Lagrangian method using a refinement close to industrial applications;
- second, to give a more complete idea of the multi-fluid method efficiency, we compare its cost to the cost of more refined DSMC Lagrangian computations.

Our interest being on combustion application, we tackle in **chapter ten** the key question, for two-phase flow combustion, of the fuel vapor mass fraction prediction. The vaporization of a polydisperse spray injected in a gaseous jet is studied. The fuel vapor mass fraction obtained by the multi-fluid is compared to the one obtained by the Lagrangian. Besides, a first simplified combustion computation, is done, also allowing to compare the methods. Finally, a 3-D multi-fluid spray computation is presented in **chapter eleven**. This represents a first achievement for the multi-fluid model, and it is validated through qualitative comparisons with a Lagrangian method, to check the description of the size-conditioned dynamics.

The present work was supported by a grant from both DGA and CNRS (Mathematics and Engineering departements, and has received the support of the INCA project (National Initiative for Advanced Combustion) led by Snecma, SAFRAN group.
Furthermore, the present research was done thanks to:

- *a Young Investigator Award from the French Ministry of Research (New Interfaces of Mathematics - M. Massot, 2003-2006);*
- *an ANR (French National Research Agency) Young Investigator Award (M. Massot, 2006-2009, N: ANR-05-JC05-42236);*
- *the support of European Commission through the project "Toward Innovative Methods for Combustion Prediction in Aero-Engines" (TIMECOP-AE, project N: AST5-CT-2006-030828).*

Part I

Polydisperse evaporating spray modeling and simulation

Introduction

The precise two-phase flow modeling framework of the study is given in this first part. The gas phase description is coupled with a kinetic spray model. This model comes from a statistical description of the dispersed liquid phase. It gives the evolution of the spray Number Density Function (NDF). This model provides a mesoscopic level of modeling; it is called kinetic by analogy with the kinetic theory of gases. Indeed, the NDF evolution equation is similar to the Boltzmann equation. The NDF evolution is driven by the physical phenomena applied to the spray. These phenomena must be described by droplet models compatible with the kinetic description. The detailed description of the kinetic modeling framework, as well as the droplet models needed for its closure, are provided in Chpt. 1. Furthermore, in order to characterize a spray resolution method, as done in this work, one needs to define a simplified modeling famework, provided in Chpt. 1. This context is the basis to conduct analysis on the spray resolution choice impact on the spray dynamics and evaporation.
The spray equation of this kinetic model cannot be resolved at the scale of industrial applications, given the associated cost. Macroscopic methods for its simulation have to be introduced. These methods can use a Lagrangian or a Eulerian approach to describe the spray. Classical Lagrangian tracking methods are first presented in Chpt. 2. Their associated level of description, as well as their link with the kinetic model, are highlighted. In the present work, Lagrangian methods are used as reference methods for the evaluation of a kinetic-based Eulerian method. It is thus essential to underline the level of description corresponding to each method. The different derivation strategies of Eulerian methods are provided in Chpt. 2. It allows to draw a classification between the different existing models, and to position the multi-fluid method adressed in this study.

Chapter 1

Kinetic modeling for dispersed liquid spray in gaseous flow

Contents

The two-phase flow modeling framework is detailed in this chapter. The present work aims at assessing a method for the dispersed-phase simulation. The spray is thus considered to be dilute, i.e., with a liquid fraction $\alpha_l = 10^{-4}$, to moderately dense, $\alpha_l = 10^{-2}$, as established by the classification done in (O'Rourke 1981). In this framework, droplets are assumed as point particles. The gas flow around and inside the droplets is not resolved and the coupling with the gas phase is done through source terms in the gas phase equations. Besides, a statistical description based on a kinetic equation for the spray NDF is considered. In this kinetic formulation, the effect of the gas on the spray can only depend of local gas properties. Long-distance interactions between gas and liquid are thus neglected. The governing equations for the gas and liquid phase are given in this chapter. The droplet models needed to close the evolution of the spray NDF are provided. Finally, a simplified modeling framework is introduced. This framework is needed to conduct precise evaluation of spray methods, as done in Part. II and Part. IV of this work. One has to notice that it is not a restrictive framework enforced by considered spray method limitations, but only a way to isolate a specific issue. As our interest is in combustion applications, an original simplified two-phase combustion model is provided.

1.1 Gaseous phase description

For the gas description, the Navier Stokes equations for a multi-species reactive flow are considered. Two strategies can be envisioned for their derivation. These Eulerian equations can be obtained through mass, momentum and energy balances on a control volume. This derivation can be found for example, for a general mono-species non-reactive case in (Candel 1990), and in (Poinsot and Veynante 2005) for reactive multi-species flows. Another, more mathematical, way to obtain these equations consists in deriving this fluid level of description from the kinetic theory of gases, (Giovangigli 1999b; Ern and Giovangigli 1994; Massot 1996). It leads to a mixt hyperbolic-parabolic system, its mathematical structure being studied in (Giovangigli and Massot 1998).
These gaseous phase equations are given in the framework of two-phase flow, with a dispersed liquid phase. This liquid phase is assumed to be dilute enough so that its influence on the gas phase can be described by source term addition in the gas phase equations. We decide to clearly explicit the non dimensional equation derivation, to underline the reference quantities arising. These quantities will also drive liquid dimensionless equation derivation, in Sec. 1.2.

1.1.1 Gaseous phase equations

The conservative form of the Navier Stokes system describes mass, species, momentum and energy conservation. The influence of the liquid phase is taken into account through mass, momentum and energy source terms.

Mass and species conservation

The conservation of mass is given by:

$$\partial_t \rho_g + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g}) = S^m, \tag{1.1}$$

where ρ_g is the gas density, $\boldsymbol{U_g}$ the velocity and S^m the mass source term from the liquid phase evaporation.

Regarding species conservation, we have, for the k^{th} species:

$$\partial_t (\rho_g Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} Y_k) = -\partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_k^{diff}} Y_k) + \dot{\omega}_k + S_k^{species}, \tag{1.2}$$

where Y_k, $\boldsymbol{U_k^{diff}}$, and $S_k^{species}$ are the mass fraction, the diffusion velocity, and the liquid evaporation mass source term of species k, respectively. The term $\dot{\omega}_k$ is the reactive rate of species k. By definition, one has:

$$Y_k = \frac{\rho_k}{\rho_g}, \qquad \rho_g \boldsymbol{U_g} = \sum_{k=1}^{n_s} \rho_g \boldsymbol{U_k} Y_k, \qquad \boldsymbol{U_k^{diff}} = \boldsymbol{U_k} - \boldsymbol{U_g}, \tag{1.3}$$

where ρ_k and $\boldsymbol{U_k}$ are the density and the hydrodynamic velocity of species k, respectively, and n_s the number of species. It yields:

$$\sum_{k=1}^{n_s} Y_k = 1, \qquad \sum_{k=1}^{n_s} \rho_k \boldsymbol{U_k^{diff}} = 0, \qquad \sum_{k=1}^{n_s} S_k^{species} = S^m. \tag{1.4}$$

To obtain the species diffusion velocity $\boldsymbol{U_k^{diff}}$, with an approximation neglecting the Soret effect, i.e., the diffusion of mass due to temperature gradients, one needs to solve a linear system of size n_s^2, see (Poinsot and Veynante 2005), in each direction at each point and for each time. This resolution being difficult and costly, (Ern and Giovangigli 1994), we use a Fick's law to compute the diffusion velocity:

$$\boldsymbol{U_k^{diff}} Y_k = -D_k \partial_{\boldsymbol{x}} \cdot (Y_k), \tag{1.5}$$

where D_k is the diffusion coefficient of species k into the mixture. The species conservation law thus becomes:

$$\partial_t (\rho_g Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} Y_k) = \partial_{\boldsymbol{x}} \cdot (\rho_g D_k \partial_{\boldsymbol{x}} \cdot (Y_k)) + \dot{\omega}_k + S_k^{species}. \tag{1.6}$$

One can notice that the first expression of Eq. (1.4), together with Eq. (1.6), leads to $n_s + 1$ equations for n_s unknowns, the system is thus over determined. Two methods can then be considered. First one can solve the species conservation Eq. (1.6) for $n_s - 1$ species and obtain the last species mass fraction Y_{n_s} using the first expression of Eq. (1.4). Second one can solve Eq. (1.6) for all the species and the first expression of Eq. (1.4) must be

obtained as a computational result. Nevertheless, as explained in (Poinsot and Veynante 2005), the introduction of Fick's law makes these two approaches not equivalent. Actually, Fick's law does not generally ensure global mass conservation, but only if the diffusion coefficients for all the species are equal, $D_k = D$, (Giovangigli 1999b). Therefore, in the first approach where global mass conservation is enforced, the mass fraction of the last species, Y_{n_s}, absorbs all the inconsistencies introduced by Fick's law in the case of different species diffusion coefficients. On the other hand, in the second approach a correction velocity must be added to the convection velocity $\boldsymbol{U_g}$, so that the global mass is conserved . It leads to the modified species conservation equation:

$$\partial_t \left(\rho_g \, Y_k\right) + \partial_{\boldsymbol{x}} \cdot \left(\rho_g \left(\boldsymbol{U_g} + \boldsymbol{U}_{\boldsymbol{g}}^{\mathrm{cor}}\right) Y_k\right) = \partial_{\boldsymbol{x}} \cdot \left(\rho_g \, D_k \partial_{\boldsymbol{x}} \cdot (Y_k)\right) + \dot{\omega}_k + S_k^{species}, \tag{1.7}$$

with

$$\boldsymbol{U}_{\boldsymbol{g}}^{\mathrm{cor}} = \sum_{k=1}^{n_s} D_k \partial_{\boldsymbol{x}} \cdot (Y_k). \tag{1.8}$$

One can refer to (Poinsot and Veynante 2005) for more details. As shown in (Ern and Giovangigli 1994), this approach is a first order approximation to the exact resolution of diffusion velocity.

Momentum conservation

The momentum conservation is given by:

$$\partial_t(\rho_g \, \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho \boldsymbol{U_g} \otimes \boldsymbol{U_g}) = \partial_{\boldsymbol{x}} \cdot (\boldsymbol{T}) + S^{mom}, \tag{1.9}$$

with $\boldsymbol{T}$ the stress tensor. This tensor can be written:

$$\boldsymbol{T} = -P_g \, \mathrm{Id} + \boldsymbol{\tau}, \tag{1.10}$$

where $P_g(\boldsymbol{x}, t)$ is the isotropic pressure field, and $\boldsymbol{\tau}$ is the viscous stress tensor. We consider, for the Navier Stokes system of equations, Newtonian fluids, i.e, with a linear isotropic behavior. In the case where volume viscosity is neglected, the stress tensor can be written, see (Candel 1990):

$$\boldsymbol{\tau} = \mu_g \left(\partial_{\boldsymbol{x}}(\boldsymbol{U_g}) + \ \partial_{\boldsymbol{x}}(\boldsymbol{U_g})^t\right) - \frac{2}{3}\mu_g \partial_{\boldsymbol{x}} \cdot (\boldsymbol{U_g}). \tag{1.11}$$

The Navier Stokes momentum equation can thus be written:

$$\partial_t(\rho_g \, \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho \boldsymbol{U_g} \otimes \boldsymbol{U_g}) \ = \ -\partial_{\boldsymbol{x}}(P_g) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}) + S^{mom}. \ . \tag{1.12}$$

Energy conservation

We consider here, amount the numerous forms of energy conservation equation, an equation for sensible enthalpy. We choose sensible enthalpy, h_s in order to eliminate the chemical term due to the formation enthalpy of species:

$$h_g = \underbrace{\int_{T_0}^{T_g} C_{p,g} \, \mathrm{d}T}_{\text{sensible}} + \underbrace{\sum_{k=1}^{n_s} \Delta h_k^0 Y_k}_{\text{chemical}}. \tag{1.13}$$

The equation for the enthalpy can be deduced from the conservation for total energy given by:

$$\partial_t(\rho_g\, e_t) + \partial_{\boldsymbol{x}} \cdot (\rho_g\, \boldsymbol{U_g}\; e_t) = -\partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{T} \cdot \boldsymbol{U_g}) + S^{energy}. \tag{1.14}$$

The variation of the total energy of the gas is due to the power produced by the stress, pressure and viscous stress, to the energy flux $\boldsymbol{q}$ and to the source term from the dispersed liquid phase, S^{energy}. The energy flux is given by:

$$\boldsymbol{q} = -\lambda_g \partial_{\boldsymbol{x}}\; T_g + \rho_g \sum_{k=1}^{n_s} h_k\; \boldsymbol{U_k^{diff}}\; Y_k. \tag{1.15}$$

It includes a heat diffusion term modeled by a Fourier's law and a term associated with different-enthalpy species diffusion. In the framework of the Fick's law assumption for the diffusion velocity we introduce the correction energy, $\boldsymbol{E}^{\rm cor}$, by analogy with the correction velocity $\boldsymbol{U_g^{\rm cor}}$ introduced for the species conservation:

$$\boldsymbol{q} = -\lambda_g \partial_{\boldsymbol{x}}\; T_g - \boldsymbol{E}^{\rm cor}, \quad \text{with} \quad \boldsymbol{E}^{\rm cor} = -\rho_g \sum_{k=1}^{n_s} h_k\; D_k\, \partial_{\boldsymbol{x}} \cdot (Y_k). \tag{1.16}$$

To obtain the enthalpy equation, we first obtain the equation for the intern energy, substracting the equation for the kinetic energy, obtained from the momentum equation. We use afterward the relation between enthalpy and energy: $h_g = e_t + P_g/\rho_g$, and we remove the formation enthalpy term: $h_s = h_g - \sum_{k=1}^{n_s} \Delta h_k^0 Y_k$. It gives, see for example (Poinsot and Veynante 2005):

$$\partial_t(\rho_g\, h_s) + \partial_{\boldsymbol{x}} \cdot (\rho_g\, \boldsymbol{U_g}\; h_s) = -\partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}) + \partial_t(P_g) + \boldsymbol{\tau} : \partial_{\boldsymbol{x}}(\boldsymbol{U_g}) + \dot{\omega}_T + S^{enth}, \tag{1.17}$$

where $\dot{\omega}_T$ is the heat release due to reaction:

$$\dot{\omega}_T = -\sum_{k=1}^{n_s} \Delta h_k^0 Y_k. \tag{1.18}$$

Finally, the gaseous phase evolution is driven by:

$$\begin{aligned}
\partial_t\, \rho_g \; + \partial_{\boldsymbol{x}} \cdot (\rho_g\, \boldsymbol{U_g}) &= S^m, \\
\partial_t\, (\rho_g\, Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g\, (\boldsymbol{U_g} + \boldsymbol{U_g^{\rm cor}}) Y_k) &= \partial_{\boldsymbol{x}} \cdot (\rho_g\, D_k \partial_{\boldsymbol{x}} \cdot (Y_k)) + \dot{\omega}_k + S_k^{species}, \\
\partial_t(\rho_g\, \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho \boldsymbol{U_g} \otimes \boldsymbol{U_g}) &= -\partial_{\boldsymbol{x}}(P_g) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}) + S^{mom}, \\
\partial_t(\rho_g\, h_s) + \partial_{\boldsymbol{x}} \cdot (\rho_g\, \boldsymbol{U_g}\; h_s) &= -\partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}) + \partial_t(P_g) + \boldsymbol{\tau} : \partial_{\boldsymbol{x}}(\boldsymbol{U_g}) + \dot{\omega}_T + S^{enth}.
\end{aligned} \tag{1.19}$$

1.1.2 Reference quantities and dimensionless formulation

In order to introduce the non dimensional equations, we define reference velocity and length based on the macroscopic characteristics of the computational domain, U_0 and x_0, allowing 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 $P_\infty = \rho_\infty R T_\infty / W_\infty$. The 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:

$$\mathrm{Re}_0 = \frac{\rho_\infty \; x_0 \; U_0}{\mu_\infty}. \tag{1.20}$$

The system is driven by the dimensionless numbers:

$$\mathrm{Sc}_k = \frac{\mu_g}{\rho_g \, D_k}, \quad \mathrm{Pr} = \frac{C_{p,g}\mu_g}{\lambda_g}, \quad \mathrm{Le}_k = \frac{\lambda_g}{\rho_g \;\; C_{p,g} \;\; D_k}. \tag{1.21}$$

The Schmidt number Sc_k for the species k, compares the mass diffusivity D_k of the species to the mechanical diffusivity μ_g/ρ_g , whereas the Prandtl number Pr compares the thermal diffusivity $\lambda_g/\rho_g \, C_{p,g}$ to the mechanical diffusivity. 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, close to the original one, we introduce the quantities:

$$\mu_g^\star = \frac{\overline{\mu_g}}{\mathrm{Re}_0}, \quad D_k^\star = \frac{\overline{\mu_g}}{\overline{\rho_g} \; \mathrm{Re}_0 \; \mathrm{Sc}_k}, \quad \lambda_g^\star = \frac{\overline{C_{p,g}} \; \overline{\mu_g}}{\mathrm{Re}_0 \; \mathrm{Pr}}. \tag{1.22}$$

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

$$\mathrm{Sc}_k = \frac{\mu_g^\star}{\overline{\rho_g} \; D_k^\star}, \quad \mathrm{Pr} = \frac{\overline{C_{p,g}} \; \mu_g^\star}{\lambda_g^\star}, \quad \mathrm{Le}_k = \frac{\lambda_g^\star}{\overline{\rho_g} \; \overline{C_{p,g}} \; D_k^\star}. \tag{1.23}$$

In this framework, we can write the gaseous system in non dimensional form, obtained from Eq. (1.19) through

$$\text{Eq. (1.25)} = \begin{pmatrix} \dfrac{x_0}{\rho_\infty \; U_0} \\ \dfrac{x_0}{\rho_\infty \; U_0} \\ \dfrac{x_0}{\rho_\infty \; U_0^2} \\ \dfrac{x_0}{\rho_\infty \; U_0^3} \end{pmatrix} \text{Eq. (1.19)}. \tag{1.24}$$

For the sake of simplicity we use for dimensionless variables, the same variable as for original ones:

$$\begin{aligned}
\partial_t \, \rho_g \; + \partial_{\boldsymbol{x}} \cdot (\rho_g \, \boldsymbol{U_g}) &= S^m, \\
\partial_t \, (\rho_g \, Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g \, (\boldsymbol{U_g} + \boldsymbol{U_g^{\mathrm{cor}}}) Y_k) &= \partial_{\boldsymbol{x}} \cdot (\rho_g \, D_k^\star \, \partial_{\boldsymbol{x}} \cdot (Y_k)) + \dot{\omega}_k + S_k^{species}, \\
\partial_t (\rho_g \, \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho_g \, \boldsymbol{U_g} \otimes \boldsymbol{U_g}) &= -\frac{1}{\gamma_\infty M^2} \partial_{\boldsymbol{x}} (P_g) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}) + S^{mom}, \\
\partial_t (\rho_g \, h_s) + \partial_{\boldsymbol{x}} \cdot (\rho_g \, \boldsymbol{U_g} \; h_s) &= -\partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}^\star) + \frac{\gamma_\infty - 1}{\gamma_\infty} \; \partial_t (P_g) \\
&\quad + M^2 (\gamma_\infty - 1) \; \boldsymbol{\tau} : \partial_{\boldsymbol{x}} \cdot (\boldsymbol{U_g}) + \dot{\omega}_T + S^{enth},
\end{aligned}$$

(1.25)

with

$$\boldsymbol{q}^\star = -\lambda_g^\star \, \partial_{\boldsymbol{x}} \, T_g + \rho_g \sum_{k=1}^{n_s} h_k \, D_k^\star \, \partial_{\boldsymbol{x}} \cdot (Y_k). \tag{1.26}$$

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

$$M = \frac{U_0}{\sqrt{\gamma_\infty \frac{R}{W_\infty} T_\infty}}, \quad \text{with} \quad \gamma_\infty = \frac{C_{p,\infty}}{C_{v,\infty}}. \tag{1.27}$$

1.2 Kinetic Spray modeling

The dispersed-phase assumption for the liquid, conducted for dilute spray with volume fraction α_l verifying $10^{-4} < \alpha_l < 10^{-2}$, allows to use the so-called spray "kinetic" model. This model is based on the Williams equation, (Williams 1958). Although it represents a mesoscopic level of description, it is called "kinetic" by analogy with the microscopic kinetic theory of gases. Indeed, the assumptions made for the spray droplets, (Fox 2007), are similar to the ones made for the gas molecules when deriving the kinetic model. These assumptions are recalled here. The model describes the transport of the NDF and its evolution due to the physical phenomena applied to the spray, as external forces, evaporation, heat transfer or particle interactions. The source terms responsible for this phase space evolution of the NDF are analyzed here. The models needed for their closure will be discussed in Sec. 1.3. Due to the dispersed-phase assumption, the liquid influence on the gas is obtained through source terms in the gas phase equations. The expressions of such source terms in the framework of kinetic spray description are provided.

1.2.1 Williams "kinetic" equation for spray

We present here the framework allowing to derive Williams "kinetic" equation for spray, see for example (Williams 1958; Williams 1985).

Basis assumptions

We recall that we focus here on a dispersed liquid phase, i.e., a spray constituted of isolated droplets. We consider that primary break-up already occur. This model needs thus, in order to simulate a full injection process in automotive or aeronautics combustion chamber, to be coupled with a model for separated two-phase flow computing the primary break-up and giving the initial conditions of the spray, i.e., droplet sizes and velocities. These initial conditions of the spray can also be obtained by experimental results or modeled, (Babinsky and Sojka 2002). This issue is essential in the framework of full injection computations. In order to describe completely the spray, we need to compute the trajectory of each particle, solving dynamical equation. It would then be necessary to take into account all the possible interactions of the particle with the gas and with the other particles. If we consider the

size and velocity evolution of one droplet, the phase space is of dimension $2N \times d$ where N is the number of droplets in the spray and d is the physical space dimension. In many applications, the order of magnitude of N is 10^9, preventing to realize such computations.

We thus adopt a statistical point of view, introducing a multiple-particle joint PDF $f_N(t, x_1, x_2, ...x_N, u_1, ...u_N)$ defined from the ensemble of all realizations of the spray. We further assume droplet independence, that can be linked with the molecular chaos assumption in the kinetic theory of gases. In this framework, one has:

$$f_N(t, x_1, x_2, ...x_N, u_1, ...u_N) = f_1(t, x_1, u_1) f_1(t, x_2, u_2) ... f_1(t, x_N, u_N), \tag{1.28}$$

where f_1 is the single-particle joint PDF. We thus only need to compute the single-particle joint PDF, $f_1(t, x_1, u_1)$. We finally define the number density function (NDF), $f(t, x, u)$, as the sum over all the spray droplets of the single-particle joint PDF. We then want to solve the evolution of the NDF of the spray.

Equation

The droplets, considered as spherical, are characterized by their position, x, their size ϕ, their velocity u and their temperature T. The number density function thus depends on these variables and on time, the quantity $f^{\phi}(t, x, \phi, u, T)\mathrm{d}t\,\mathrm{d}x\,\mathrm{d}\phi\,\mathrm{d}u\,\mathrm{d}T$ being the probable number of droplets with, at time t, a position in $[x, x + \mathrm{d}x]$, a size in $[\phi, \phi + \mathrm{d}\phi]$, a velocity in $[u, u + \mathrm{d}u]$ and a temperature in $[T, T + \mathrm{d}T]$. The particle size can be described either by their volume V, surface S or radius R, with the following relation:

$$f^S \mathrm{d}S = f^R \mathrm{d}R = f^V \mathrm{d}V. \tag{1.29}$$

It was shown in (Laurent 2006), that taking the droplet surface as size variable is a good choice to develop multi-fluid model from the kinetic description. We therefore use surface S as the size variable in the following. The number density function f, without any superscript, will then be associated to the droplet surface. The NDF function of the spray follows a transport equation, first introduced in (Williams 1958), similar to a Boltzmann equation:

$$\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{u} f) + \partial_{\boldsymbol{u}} \cdot (\boldsymbol{F}\, f) - \partial_S \left(R_s\ f\right) + \partial_T \left(E f\right) = \Gamma + Q, \tag{1.30}$$

where:

- $\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{u} f)$ represents the free transport of the spray;
- $\boldsymbol{F} = \mathrm{d}_t(\boldsymbol{u})$ is the force applied on droplets per unit mass;
- $R_s = -\mathrm{d}_t(S)$ is the rate of change of the size S of droplets;
- $E = \mathrm{d}_t(T)$ is the rate of change of droplet 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 through particle formation by secondary break-up process.

One can note the level of description of the kinetic model of the dispersed phase: the gas influence on the spray is modeled through the terms $\boldsymbol{F}$, R_s and E. These terms thus depend on the gas temperature, velocity and composition at the position of the droplet, and therefore they depend on space and time.

1.2.2 Source terms for spray NDF evolution

The expressions of the terms $\boldsymbol{F}$, R_s, E, Γ and Q are closed by droplet models. The physical phenomena represented by each term are first detailed before entering in the model choice details. The models considered here are single isolated mono-component droplet models. The reader can find, for instance, in (Sirignano 1999), models for heating and vaporization of droplet groups or for multicomponent liquid droplets.

Forces per unit mass $\boldsymbol{F}$ We can decompose this term into four contributions, see for example (Crowe, Sommerfeld, and Tsuji 1998):

$$\boldsymbol{F} = \frac{1}{m_p}\left(\boldsymbol{F}_{\mathrm{G}} + \boldsymbol{F}_{\mathrm{D}} + \boldsymbol{F}_{\mathrm{vm}} + \boldsymbol{F}_{\mathrm{Basset}} + \boldsymbol{F}_{\mathrm{L}}\right), \tag{1.31}$$

where m_p is the mass of a droplet, $m_p = \rho_l S^{3/2}/\left(6\sqrt{\pi}\right)$.

- $\boldsymbol{F}_{\mathrm{G}}$ contains gravity with buoyancy effects, known as Archimedes principle. Details on buoyancy effects can be found for example in (Sirignano 1999).
- $\boldsymbol{F}_{\mathrm{D}}$ is the drag force.
- $\boldsymbol{F}_{\mathrm{vm}}$ and $\boldsymbol{F}_{\mathrm{Basset}}$, the virtual or apparent mass effect and the Basset force, are due to droplet acceleration. Indeed, the relative velocity acceleration induces two effects. The first one, described by the virtual mass effect is related to drag and corresponds to the acceleration of the gas carried by the particle. The second one, described by the Basset force, is related to the viscous effects. This term addresses the temporal delay in the boundary layer development as the relative velocity changes with time. This boundary layer development has then an effect on the droplets. These terms are not compatible with the kinetic spray description, see (Laurent 2002b).
- $\boldsymbol{F}_{\mathrm{L}}$ represents the lift force, due to particle rotation.

As shown in (Dufour 2005), in a gas-liquid flow, where ρ_g/ρ_l is of the order of 10^{-2} down to 10^{-3}, the only external forces that need to be accounted for are drag force and gravity. Furthermore, in order to work in a simple modeling framework, we neglected gravity in the studies presented in this work, except in Chpt. 5 for the comparisons with experimental measurements.

Droplet size change rate R_s The droplet size can decrease due with evaporation or increase with dilatation. Dilatation is due to a variable mass density:

$$\mathrm{d}_t(\rho_l\, Vol) = \underbrace{\rho_l\, \mathrm{d}_t(Vol)}_{\text{evaporation}} + \underbrace{Vol\, \mathrm{d}_t(\rho_l)}_{\text{dilatation}}, \tag{1.32}$$

Vol being the Volume of the droplet, given by $Vol(S) = S^{3/2}/(6\sqrt{\pi})$. The mass density evolution is mainly due to evaporation. We consider in this work, a constant mass density for the liquid. The physical models used to describe droplet size evolution are given in Sec. 1.3.2.

Droplet temperature change rate E The droplet temperature evolves due to convective and radiative heating. The main part of droplet heating is convective, nevertheless in combustion application fuel droplets can be heated by radiation from high-temperature gases. We restrict ourselves here to conductive heating. The reader can consult (Sirignano 1999) and references therein for more details about modeling radiative heating of droplets. Furthermore, we assume a uniform temperature T in the droplet and thus we neglect heat conduction inside the droplet. The models used in this framework are detailed in Sec. 1.3.2.

Source terms Γ **and** Q These terms represent droplet-droplet collisions, and secondary break-up phenomena, respectively. We will present here only a brief summary of the modeling for these phenomena, as they will not be accounted for in the studies presented in this work. We refer to (Villedieu and Hylkema 1997; Laurent, Massot, and Villedieu 2004) for the details.

1.2.3 Spray source terms for gas resolution

The kinetic spray modeling derived above, allows to compute the source terms arising in the gaseous equations, Eq. (1.19), taking into account the influence of the dispersed liquid phase on the carrier gaseous phase. 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. For genericity purpose, the terms are written for any size variable ϕ.

$$\begin{aligned}
S^m &= \iiint \rho_l \, R_\phi \, \mathrm{d}_\phi(Vol) \, f^\phi \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \\
S_k^{species} &= \iiint \Omega_\mathrm{k}^\mathrm{Surf} \, \rho_l \, R_\phi \, \mathrm{d}_\phi(Vol) \, f^\phi \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \\
S^{mom} &= \iiint \rho_l \, \boldsymbol{u} \, R_\phi \, \mathrm{d}_\phi(Vol) \, f^\phi \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T + \iiint \rho_l \, Vol \, \boldsymbol{F} \, f^\phi \, \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \\
S^{enth} &= \iiint \rho_l \, C_{p,l} \, T \, R_\phi \, \mathrm{d}_\phi(Vol) \, f^\phi \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T \iiint \rho_l \, Vol \, C_{p,l} \, E \, \hat{\mathrm{A}} \, f^\phi \, \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T,
\end{aligned} \tag{1.33}$$

where $Vol(\phi)$ and R_ϕ are the volume and the size change rate corresponding to the size ϕ, respectively. Moreover, $\Omega_\mathrm{k}^\mathrm{Surf}$ is the flux fraction for species k, from liquid surface to gas. If the droplets are mono-component, that will be the case in all our study, $\Omega_\mathrm{k}^\mathrm{Surf} = 0$ or $\Omega_\mathrm{k}^\mathrm{Surf} = 1$. 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.

The two-way coupling is illustrated here, the gas influence over the spray resulting in F, R_ϕ and E 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. (1.33). The computation of these source terms in the multi-fluid model framework is detailed in Chpt. 2.

1.3 Droplet models for NDF evolution closure

The evolution of the spray NDF in the phase space $(\boldsymbol{u}, S, T)$, is considered here. The source terms in the spray equation leading to this evolution and representing physical phenomena are based on classical models for isolated droplets. The droplet models compatible with the kinetic description of the spray are provided here. Drag force, heat transfer and droplet interactions description are considered.

1.3.1 Drag Force

The general expression for the drag force is given by, (O'Rourke 1981; Sirignano 1999):

$$\boldsymbol{F}_{\mathrm{D}} = \frac{1}{8}\,\rho_g\, C_D\, S\, (\boldsymbol{U_g} - \boldsymbol{u})\, ||\boldsymbol{U_g} - \boldsymbol{u}||, \tag{1.34}$$

where C_D is the drag coefficient. This coefficient depends on the particle shape as well as on the flow parameters such as Reynolds number, Mach number, turbulence, etc. In the kinetic model we assume the particles spherical, and we focus only on the variation of C_D with the Reynolds number based on the relative velocity:

$$\mathrm{Re_p} = \frac{\rho_g\, S^{1/2} ||\boldsymbol{U_g} - \boldsymbol{u}||}{\sqrt{\pi}\mu_g}. \tag{1.35}$$

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

$$C_D = \frac{24}{\mathrm{Re_p}}, \quad \mathrm{Re_p} < 1. \tag{1.36}$$

In this case the drag force term can be written:

$$\boldsymbol{F}_{\mathrm{D}} = m_p\, \frac{18\,\pi\,\mu_g}{\rho_l\, S}\, (\boldsymbol{U_g} - \boldsymbol{u}) = m_p\, \frac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p}, \tag{1.37}$$

where

$$\tau_p = \frac{\rho_l\, S}{18\,\pi\,\mu_g}, \tag{1.38}$$

is the momentum or velocity response time. The variable $\tau_p(S)$ represents the time required for a droplet of size S to respond to a change in gas velocity. This simplified form of the drag coefficient is only correct for $\mathrm{Re_p} < 1$. To cover the rest of the Stokes flow regime, we use the experimental correlation proposed by Schiller and Naumann, see (Schiller and Naumann 1935) or (Clift, Grace, and Weber 1978):

$$C_D = \frac{24}{\mathrm{Re_p}} \left(1 + \frac{\mathrm{Re_p^{2/3}}}{6}\right), \quad \mathrm{Re_p} < 1000. \tag{1.39}$$

For the inertial range regime we can consider, see (Crowe, Sommerfeld, and Tsuji 1998):

$$C_D = 0.45, \quad 1000 < \mathrm{Re_p} < 3.5 \times 10^5. \tag{1.40}$$

We have thus, for the drag force:

$$\boldsymbol{F}_\mathrm{D} = \begin{cases} m_p \dfrac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p}, & \mathrm{Re_p} < 1 \\ m_p \dfrac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p} \left(1 + \dfrac{\mathrm{Re_p^{2/3}}}{6}\right), & 1 < \mathrm{Re_p} < 1000 \\ m_p \dfrac{3\,\rho_g\,\sqrt{\pi}}{4\,\rho_l\,S^{1/2}}\, 0.45\, (\boldsymbol{U_g} - \boldsymbol{u})\, ||\boldsymbol{U_g} - \boldsymbol{u}||, & 10^3 < \mathrm{Re_p} < 3.5 \times 10^5. \end{cases} \tag{1.41}$$

If the forces taken into account reduce to the drag force, as in most of the cases of this work, then the force per unit mass is given by:

$$\boldsymbol{F} = \begin{cases} \dfrac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p}, & \mathrm{Re_p} < 1 \\ \dfrac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p} \left(1 + \dfrac{\mathrm{Re_p^{2/3}}}{6}\right), & 1 < \mathrm{Re_p} < 1000 \\ \dfrac{3\,\rho_g\,\sqrt{\pi}}{4\,\rho_l\,S^{1/2}}\, 0.45\, (\boldsymbol{U_g} - \boldsymbol{u})\, ||\boldsymbol{U_g} - \boldsymbol{u}||, & 10^3 < \mathrm{Re_p} < 3.5 \times 10^5. \end{cases} \tag{1.42}$$

Furthermore, in order to evaluate new numerical methods in a simple modeling framework, we reduce to Stokes law for most of the studies presented in this work.

1.3.2 Evaporation and Heat Transfer

We present in this section the model needed to obtain the size and temperature change rate R_s and E. We first write their general formulation before presenting the models chosen for closure. We recall here the main steps of the models, more details can be found in (Versaevel 1996; Abramzon and Sirignano 1989).

Size and temperature evolution terms

Size change rate The change rate of the size is linked to droplet mass variation and thus to the total mass flux of vapor at the droplet surface, $[\dot{m}_p]_s$:

$$[\dot{m}_p]_s = -\mathrm{d}_t \left(\frac{\rho_l\, S^{3/2}}{6\,\sqrt{\pi}}\right) = -\rho_l\, \frac{S^{1/2}}{4\,\sqrt{\pi}} \mathrm{d}_t S = \rho_l\, \frac{S^{1/2}}{4\,\sqrt{\pi}} R_s. \tag{1.43}$$

As previously mentioned, we neglect in Eq. (1.43) the dilatation of the droplet, so that $\rho_l = \textit{constant}$. We therefore need to compute the vapor mass flux to obtain the R_s expression.

To do so we consider an isolated spherical droplet. In order to take into account convection cases, i.e., with a relative velocity between the droplet and the gas, we define a diffusive

mass boundary layer, δ_m. Beyond this distance, the mass vapor mass fraction is taken as a constant: $Y_F(\delta_m) = [Y_F]_\infty$. If there is no relative velocity, $\delta_m \to \infty$. We also introduce the convective Sherwood number, $\mathrm{Sh_c}$:

$$\mathrm{Sh_c} = 2\,\frac{\delta_m}{\delta_m - R}. \tag{1.44}$$

We take advantage of the spherical symmetry to derive a 1-D formulation, using spherical coordinates with the droplet center as origin. We thus use the radius R of the droplet as its size variable. In this framework, the flux $[\dot{m}_p]_s$ can be written, see (Reveillon 2007):

$$[\dot{m}_p]_s = 4\pi R^2\,\rho_g\,[\boldsymbol{U_g}]_s, \tag{1.45}$$

where $[\boldsymbol{U_g}]_s$ is the gas velocity at the droplet surface. We thus need to evaluate the quantity $\rho_g\,[\boldsymbol{U_g}]_s$, to obtain the size rate expression.

To do so, we consider the conservation equation for the fuel vapor mass fraction, Y_F, and we assume that the evaporation process is slow enough to consider a steady state. We also assume a Fick's law for the fuel vapor diffusion in the surrounding gas. Finally, we consider the limit of no chemical reaction, see (Sirignano 1999). In the 1-D spherical symmetry framework, it takes the form:

$$\frac{1}{r^2}\mathrm{d}_r\left(\rho_g\,U_g r^2\,\mathrm{d}_r(Y_F)\right) = \mathrm{d}_r\left(r^2\,\rho_g\;D_{Y_F}\,\mathrm{d}_r(Y_F)\right), \tag{1.46}$$

where D_{Y_F} is the Fick's law binary diffusion coefficient. Furthermore, the mass conservation in the 1-D spherical formulation leads:

$$\frac{1}{r^2}\mathrm{d}_r\left(\rho_g\,U_g r^2\right) = 0, \tag{1.47}$$

and therefore, $\rho_g\,U_g r^2 = \mathit{constant}$. The vapor fraction conservation becomes:

$$\rho_g\,U_g r^2\,\mathrm{d}_r(Y_F) = \mathrm{d}_r\left(r^2\,\rho_g\;D_{Y_F}\,\mathrm{d}_r(Y_F)\right), \tag{1.48}$$

In order to obtain $\rho_g\,[\boldsymbol{U_g}]_s$, we will integrate this equation within the diffusive boundary layer. First of all we rewrite Eq. (1.48), introducing the transfer parameter b:

$$\rho_g\,U_g r^2\,\mathrm{d}_r(\mathrm{b}) = \mathrm{d}_r\left(r^2\,\rho_g\;D\,\mathrm{d}_r(\mathrm{b})\right), \qquad \mathrm{b} = \frac{Y_F(r)}{([Y_F]_s - 1)}. \tag{1.49}$$

We then integrate between the droplet radius $r = R$, to the end of the diffusive mass boundary layer $r = \delta_m$, where $\mathrm{b} = \mathrm{b}_\infty$. This integration leads to, see (Reveillon 2007):

$$R\,[\boldsymbol{U_g}]_s = \frac{1}{2}\,\mathrm{Sh_c}\,D_{Y_F}\ln(1+\mathrm{B_M}), \tag{1.50}$$

where we introduced the Spalding dimensionless mass transfer number:

$$\mathrm{B_M} = \mathrm{b}_\infty - [\mathrm{b}]_s = \frac{[Y_F]_s - [Y_F]_\infty}{1 - [Y_F]_s}. \tag{1.51}$$

Eq. (1.50) gives:

$$[\dot{m}_p]_s = 2\pi R\,\rho_g\;\mathrm{Sh_c}\,D_{Y_F}\ln(1+\mathrm{B_M}), \tag{1.52}$$

and combined with Eq. (1.43), it gives the expression for R_s:

$$R_s = 4\pi\,\frac{\rho_g}{\rho_l}\,\mathrm{Sh_c}\,D_{Y_F}\ln(1+\mathrm{B_M}). \tag{1.53}$$

The $\mathrm{Sh_c}$ and $\mathrm{B_M}$ coefficients will depend on the model choice, discussed afterward.

Temperature change rate The droplet heating by the gas is used by part to vaporize the droplet and by part to heat the liquid core. The conductive heat reaching the droplet surface may then be written, see (Reveillon 2007):

$$Q_g = \underbrace{\dot{m}_p L_v}_{\text{vaporization}} + \underbrace{Q_l}_{\text{heating}}, \tag{1.54}$$

L_v being the latent heat of vaporization, Q_l the heat absorbed by the liquid core. We recall that only thermal conduction is accounted for, therefore Q_g is given by a conductive flux, possibly modified by convective effects:

$$Q_g = 4\pi R^2 \lambda_g \, \mathrm{d}_t T_g = 4\pi R^2 \, h_c (T_\infty - [T]_s), \tag{1.55}$$

where h_c is the convective modified heat transfer coefficient. The heat Q_l induces the droplet temperature variation and is thus linked to the temperature change rate:

$$Q_l = m_p \, C_{p,l} \, \mathrm{d}_t(T) = m_p \, C_{p,l} \, E, \tag{1.56}$$

where we assume a constant heat capacity $C_{p,l}$ for the liquid phase. The temperature rate of change is thus given by:

$$E = \frac{1}{m_p \, C_{p,l}} \left(Q_g - \dot{m}_p L_v\right) = \frac{6\sqrt{\pi}}{\rho_l S^{3/2} C_{p,l}} \left(Q_g - \dot{m}_p L_v\right). \tag{1.57}$$

To obtain the expression of the temperature change rate, we derive a new expression for $\dot{m}_p$ integrating the energy conservation equation over the thermal diffusive boundary layer. To do so we use exactly the same 1-D spherical symmetry context as for the fuel vapor conservation. We define the thermal diffusive boundary layer δ_T, with $T_g(\delta_T) = T_\infty$. We also introduce the associated convective Nusselt number:

$$\mathrm{Nu_c} = 2 \, \frac{\delta_T}{\delta_T - R}. \tag{1.58}$$

The energy conservation equation can be written:

$$\rho_g \, U_g r^2 \, \mathrm{d}_r(C_{p,g} T_g) = \mathrm{d}_r \left(r^2 \, \lambda_g \, \mathrm{d}_r(T_g)\right), \tag{1.59}$$

and its integration leads, see (Reveillon 2007):

$$[\dot{m}_p]_s = 2\pi \, R \frac{1}{C_{p,g}} \, \mathrm{Nu_c} \, \lambda_g \ln(1 + \mathrm{B_T}), \tag{1.60}$$

where we introduce the dimensionless heat transfer number $\mathrm{B_T}$:

$$\mathrm{B_T} = \frac{\rho_g \, U_g C_{p,g}}{h_c}. \tag{1.61}$$

Finally, using Eq. (1.55), Eq. (1.56) and Eq. (1.60), we have for the temperature rate of change the following expression:

$$E = \frac{2\pi R \lambda_g}{m_p \, C_{p,l}} \left(\mathrm{Nu} \, (T_\infty - [T]_s) - \frac{L_v}{C_{p,g}} \mathrm{Nu_c} \ln(1 + \mathrm{B_T})\right), \tag{1.62}$$

with $\mathrm{Nu} = h_c\, 2R/\lambda_g$. Using $\mathrm{Nu} = \mathrm{Nu_c}\ln(1+\mathrm{B_T})/\mathrm{B_T}$, we finally have:

$$E = \frac{2\pi R\lambda_g}{m_p\, C_{p,l}\, C_{p,g}} \mathrm{Nu_c}\, \ln(1+\mathrm{B_T}) \left(\frac{C_{p,g}(T_\infty - [T]_s)}{\mathrm{B_T}} - L_v \right). \tag{1.63}$$

Using the two expressions derived for the flux $[\dot{m}_p]_s$, Eq. (1.52) and Eq. (1.60), gives:

$$E = \frac{2\pi R\, \rho_g}{m_p\, C_{p,l}} \mathrm{Sh_c}\, D_{Y_F}\, \ln(1+\mathrm{B_M}) \left(\frac{C_{p,g}(T_\infty - [T]_s)}{\mathrm{B_T}} - L_v \right), \tag{1.64}$$

and

$$\mathrm{B_T} = (\mathrm{B_M}+1)^{\frac{\mathrm{Sh_c}}{\mathrm{Nu_c}}\frac{\mathrm{Sc}}{\mathrm{Pr}}} - 1. \tag{1.65}$$

In the dispersed two-phase flow context, T_∞ is the gas temperature at the droplet position: $T_g(x_p)$. We give E expression in function of the droplet size:

$$E = \frac{6\pi\, \rho_g}{\rho_l S\, C_{p,l}} \mathrm{Sh_c}\, D_{Y_F}\, \ln(1+\mathrm{B_M}) \left(\frac{C_{p,g}(T_g(x_p) - [T]_s)}{\mathrm{B_T}} - L_v \right). \tag{1.66}$$

In this expression, $\mathrm{Sh_c}$, $\mathrm{B_M}$ and $\mathrm{B_T}$ are still to be modeled.

Model choice for closure

Models without internal circulation description are considered in this work. Indeed the temperature is assumed uniform within the droplet, the temperature profile is not described. There exist different models to take this profile into account, see (Abramzon and Sirignano 1989; Sirignano 1999), nevertheless, these models are not compatible with the kinetic description of the spray, as explained in (Laurent and Massot 2001). Among the model with no internal circulation, we focus in this work to constant droplet-temperature model and infinite liquid-conductivity model.

Constant droplet-temperature model A first possible model is to neglect droplet heating. The temperature rate of change $E = 0$. If this model is considered without taking convective corrections into account, it leads the D^2 law (Godsave 1953), (Spalding 1953), where the square of the droplet diameter D has a linear regression in time:

$$\mathrm{d}_t(D^2) = -K. \tag{1.67}$$

The coefficient K does not depend on the droplet diameter, but depends on local gas conditions. The absence of relative velocity yields $\delta_m \to \infty$ and thus $\mathrm{Sh_c} = 2$. The rate of change of size R_s is thus given by:

$$R_s = 8\pi \frac{\rho_g}{\rho_l} D_{Y_F} \ln(1+\mathrm{B_M}). \tag{1.68}$$

The mass transfer number $\mathrm{B_M}$ is obtained solving a saturation law, (Reveillon 2007). It requires the knowledge of the droplet surface temperature $[T]_s$. Since the temperature is

Fuel	B_M
iso-octane	6.41
benzene	5.97
n-heptane	5.82
kerosene	3.4
gazole	2.5
carbon	0.12

Table 1.1 – *Constant B_M approximation: values for different fuels, for fuel-air mixture.*

not resolved, we need to compute it. We use the fact that droplet heating is not accounted for, i.e., $E = 0$, giving, from Eq. (1.66), to:

$$\left(\frac{C_{p,g}(T_g(x_p) - [T]_s)}{\mathrm{B_T}([T]_s)}\right) = L_v([T]_s). \tag{1.69}$$

Nevertheless in many cases, the D^2 law is associated with a constant $\mathrm{B_M}$ number. This approximation stands if the temperature and pressure of the gaseous carrier phase are uniform and if the vapor concentration is far from the saturation point. We report on Table 1.3.2 the value of constant $\mathrm{B_M}$ for different fuels mixed with air, taken from (Reveillon 2007). For the applications aimed at, this method, neglecting droplet heating, is clearly not the best suited, and it is necessary to take heating into account, see for example (Abramzon and Sirignano 1989). Nevertheless, it offers a very interesting framework for model evaluation, given its simplicity and easiness of use.

Infinite liquid-conductivity model A first model taking into account droplet heating, but still not resolving internal conduction, is the infinite liquid-conductivity model. Droplet temperature is thus uniform, $[T]_s = T$, but depends on time. The rate of change of size, R_s, and of temperature, E, are given by Eq. (1.53) and Eq. (1.66), respectively. The mass transfer number $\mathrm{B_M}$ is given by a saturation law determined by the local properties of the flow, see for example (Canneviere 2003). The thermal transfer number $\mathrm{B_T}$ is obtained from $\mathrm{B_M}$, through Eq. (1.65). In order to compute $\mathrm{B_M}$ and $\mathrm{B_T}$, it remains to compute the values of the corrected Sherwood and Nusselt numbers, $\mathrm{Sh_c}$ and $\mathrm{Nu_c}$. Their values depend on convection corrections.

- If no corrections are taken into account, we have, as in the previous model:

$$\mathrm{Sh_c} = \mathrm{Nu_c} = 2. \tag{1.70}$$

  In the limit where droplet heating tends to zero, we obtain the previous D^2 law model.

- To take convection into account, corrections are inserted in $\mathrm{Sh_c}$ and $\mathrm{Nu_c}$. We present here two possible choices among several available in the literature, presented for example in (Sirignano 1999). A first choice is to take the empirical corrections form

Faeth and Fendell (Faeth 1983), leading to:

$$\begin{aligned} \mathrm{Sh_c} &= 2 + \frac{0.55 \mathrm{Re_p}\,\mathrm{Sc}}{\left(1.232 + \mathrm{Re_p}, \mathrm{Sc}^{4/3}\right)^{1/2}}, \\ \mathrm{Nu_c} &= 2 + \frac{0.55 \mathrm{Re_p}\,\mathrm{Pr}}{\left(1.232 + \mathrm{Re_p}, \mathrm{Pr}^{4/3}\right)^{1/2}}. \end{aligned} \tag{1.71}$$

- A second choice for convection correction insertion is to take the method proposed in (Abramzon and Sirignano 1989), where:

$$\begin{aligned} \mathrm{Sh_c} &= 2 + \frac{\mathrm{Sh} - 2}{\mathrm{F_M}}, \\ \mathrm{Nu_c} &= 2 + \frac{\mathrm{Nu} - 2}{\mathrm{F_T}}, \end{aligned} \tag{1.72}$$

where:

$$\begin{aligned} \mathrm{F_M} &= (1 + \mathrm{B_M})^{0.7}\, \frac{\ln(1 + \mathrm{B_M})}{\mathrm{B_M}}, \\ \mathrm{F_T} &= (1 + \mathrm{B_T})^{0.7}\, \frac{\ln(1 + \mathrm{B_T})}{\mathrm{B_T}}. \end{aligned} \tag{1.73}$$

The classical Sherwood and Nusselt number are determined using the correlation given in (Clift, Grace, and Weber 1978):

$$\begin{aligned} \mathrm{Sh} &= 1 + (1 + \mathrm{Re_p}\,\mathrm{Sc})^{1/3}\, \psi(\mathrm{Re_p}), \\ \mathrm{Nu} &= 1 + (1 + \mathrm{Re_p}\,\mathrm{Pr})^{1/3}\, \psi(\mathrm{Re_p}), \end{aligned} \tag{1.74}$$

with

$$\begin{cases} \psi(\mathrm{Re_p}) = 1 \quad \text{if} \quad \mathrm{Re_p} \leq 1, \\ \psi(\mathrm{Re_p}) = \mathrm{Re_p}^{0.077} \quad \text{if} \quad 1 < \mathrm{Re_p} < 400. \end{cases} \tag{1.75}$$

The use of these correlations requires an iterative procedure, the correction factors, $\mathrm{F_M}$ and $\mathrm{F_T}$ being function of $\mathrm{B_M}$ and $\mathrm{B_T}$, respectively.

1.3.3 Droplet interactions

Collisions

Collision kinetic modeling for spray is detailed in (Villedieu and Hylkema 1997), (Dufour 2005). Its derivation is similar to Boltzmann equation derivation for rarefied gases, see (Cercignani 1988; Cercignani, Illner, and Pulvirenti 1994). We assume binary collisions, that is verified if the spray is diluted, and we neglect the influence of the impact parameter. The collision term has two contributions : $\Gamma = \Gamma_c + \Gamma_r$, where Γ_c represents collisions resulting in coalescence, while Γ_r are collisions resulting in rebounds.

As presented in (Laurent 2002b; Laurent, Massot, and Villedieu 2004), we will model the coalescence term as $\Gamma_c = \Gamma_c^+ + \Gamma_c^-$ where

$$\begin{aligned}\Gamma_c^+ &= \frac{1}{2}\int_{V^\star}\int_{\boldsymbol{u}^\star} E_{\text{coal}}(V,V^\star)B(V^\diamond,V^\star)|\boldsymbol{u}^\diamond-\boldsymbol{u}^\star|f^V(V^\diamond,\boldsymbol{u}^\diamond)f^V(V^\star,\boldsymbol{u}^\star)J\,\mathrm{d}V^\star\mathrm{d}\boldsymbol{u}^\star,\\ \Gamma_c^- &= -\int_{V^\star}\int_{\boldsymbol{u}^\star} E_{\text{coal}}(V,V^\star)B(V,V^\star)|\boldsymbol{u}-\boldsymbol{u}^\star|f^V(V,\boldsymbol{u})f^V(V^\star,\boldsymbol{u}^\star),\mathrm{d}V^\star\mathrm{d}\boldsymbol{u}^\star.\end{aligned} \tag{1.76}$$

Thereby, Γ_c^+ represents the forming of a droplet with volume V and velocity $\boldsymbol{u}$, obtained by the coalescence between a droplet with volume $V^\star$ and velocity $\boldsymbol{u}^\star$ and a droplet with volume $V^\diamond$ and velocity $\boldsymbol{u}^\diamond$. On the other hand, Γ_c^- represents the disappearance of a droplet with volume V and velocity $\boldsymbol{u}$ due to collision with a droplet of volume $V^\star$ and velocity $\boldsymbol{u}^\star$. We thus have the following relations:

$$\begin{aligned}V^\diamond &= V - V^\star,\\ \boldsymbol{u}^\diamond &= \frac{V\boldsymbol{u}-V^\star\boldsymbol{u}^\star}{V-V^\star}.\end{aligned} \tag{1.77}$$

$J = (V/V^\diamond)^d$ is the Jacobian matrix of the transformation $(V,\boldsymbol{u}) \to (V^\diamond,\boldsymbol{u}^\diamond)$ with fixed $(V^\star,\boldsymbol{u}^\star)$, where d is the number of dimension of the velocity phase space. The coalescence probability is E_{coal}, and the collision cross section is

$$B(V,V^\star) = B(V^\star,V) = \pi\left(R(V)+R(V^\star)\right)^2, \tag{1.78}$$

where the droplet radius is $R(V) = (3V/(4\pi))^{1/3}$.

In this work, we will model the rebound term, Γ_r, using the generalized hard-sphere Boltzmann inelastic, binary-collision operator (Cercignani 1988), given for a 3-D velocity phase space:

$$\begin{aligned}\Gamma_r &= \int_0^\infty (1-E_{\text{coal}}(S,S^\star))B(S,S^\star)\\ &\quad \frac{1}{\pi}\int_{R^3}\int_{S^+}\left[\frac{f(S,\boldsymbol{u''})f(S^\star,\boldsymbol{u''})}{\alpha(S,S^\star)^2} - f(S,\boldsymbol{u})f(S^\star,\boldsymbol{u}^\star)\right]|\boldsymbol{g}\cdot\boldsymbol{n}|\,\mathrm{d}\boldsymbol{n}\mathrm{d}\boldsymbol{u}^\star\mathrm{d}S^\star,\end{aligned} \tag{1.79}$$

where $\boldsymbol{n} = (\boldsymbol{x}-\boldsymbol{x}^*)/|\boldsymbol{x}-\boldsymbol{x}^*|$ is the unit vector in the direction between the droplet centers, $\boldsymbol{g} = \boldsymbol{u}-\boldsymbol{u}^\star$ is the velocity difference before collision, and $0 \le \alpha \le 1$ is the coefficient of restitution with the property $\alpha(S,S^\star) = \alpha(S^\star,S)$. For elastic rebounds, one has $\alpha = 1$. The surface S^+ is the unit half sphere on which $\boldsymbol{g}\cdot\boldsymbol{n} > 0$ (i.e., velocity differences that result in collisions). The double-prime variables denote values before the inverse collision, which are defined in terms of the pre-collision values by (Vedula, Fox, and Boyd 2008).

$$\begin{aligned}\boldsymbol{u''} &= \boldsymbol{u} - \frac{V^\star(1+\alpha(V,V^\star))}{\alpha(V,V^\star)(V+V^\star)}(\boldsymbol{g}\cdot\boldsymbol{n})\boldsymbol{n},\\ \boldsymbol{u}^{\star\prime\prime} &= \boldsymbol{u}^\star + \frac{V(1+\alpha(V,V^\star))}{\alpha(V,V^\star)(V+V^\star)}(\boldsymbol{g}\cdot\boldsymbol{n})\boldsymbol{n}.\end{aligned} \tag{1.80}$$

The volume of the droplet is given as a function of its size: $V(S) = S^{3/2}/(6\sqrt{\pi})$ and $V^\star(S^\star) = S^{\star 3/2}/(6\sqrt{\pi})$ Note that we have assumed in Eq. (1.79) and Eq. (1.80) that the droplet volumes do not change during a rebound event (i.e. $V'' = V' = V$ and $V^{\star\prime\prime} = V^{\star\prime} = V^\star$). We present in this work, some results, obtained in (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009), where Boltzmann collisions are computed in a 2-D jet framework. This case can be seen as a 3-D case with all the droplet centers and velocity initially in the same plane. In this case, Γ_r is given by:

$$\begin{aligned}\Gamma_\mathrm{r} &= \int_0^\infty (1 - E_{\mathrm{coal}}(S, S^\star)) B_{2D}(S, S^\star) \\ &\quad \frac{1}{\pi}\int_{R^2}\int_{D^+}\left[\frac{f(S,\boldsymbol{u''})f(S^\star,\boldsymbol{u''})}{\alpha(S,S^\star)^2} - f(S,\boldsymbol{u})f(S^\star,\boldsymbol{u}^\star)\right]|\boldsymbol{g}\cdot\boldsymbol{n}|\,\mathrm{d}\boldsymbol{n}\mathrm{d}\boldsymbol{u}^\star\mathrm{d}S^\star,\end{aligned} \tag{1.81}$$

where D^+ is the unit half circle on which $\boldsymbol{g}\cdot\boldsymbol{n} > 0$. The collision cross section is given in that case by:

$$B_{2D}(S, S^\star) = B(S^\star, S) = R(S) + R(S^\star). \tag{1.82}$$

One can note that the collision term is defined such that mass and momentum are conserved:

$$\int_{R^3}\int_0^\infty \rho_l \frac{S^{3/2}}{6\sqrt{\pi}}\,\Gamma(S,\boldsymbol{u})\,\mathrm{d}S\mathrm{d}\boldsymbol{u} = 0, \tag{1.83}$$

and

$$\int_{R^3}\int_0^\infty \rho_l \frac{S^{3/2}}{6\sqrt{\pi}}\,\boldsymbol{u}\,\Gamma(S,\boldsymbol{u})\,\mathrm{d}S\;\mathrm{d}\boldsymbol{u} = 0. \tag{1.84}$$

Secondary break-up

The secondary break-up is mainly due to an important relative velocity between the gas and the droplet, the droplet experiencing in this case important pressure stress causing its deformation and eventually breakage. This phenomena can be evaluated by the dimensionless Weber number, comparing aerodynamical force, related to dynamic pressure, to the force of surface tension, σ:

$$\mathrm{We} = \frac{2R\rho_g\,||\boldsymbol{U_g} - \boldsymbol{u}||^2}{\sigma}. \tag{1.85}$$

For Weber number $\mathrm{We} > 12$, secondary break-up has to be taken into account. The details of its modeling can be found in (Achim 1999; Hylkema 1999; Villedieu and Hylkema 2000; Dufour 2005), we only recall here the key points needed to compute the break-up operator Q.

As in the coalescence case, we can decompose Q into a creation and a destruction part: $Q = Q^+ + Q^-$:

$$\begin{aligned}Q^+(R,\boldsymbol{u}) &= \int_{\Omega_v} \nu^{bup}(R^\star,\boldsymbol{u}^\star)h(R,\boldsymbol{u},R^\star,\boldsymbol{u}^\star)f^R(R^\star,\boldsymbol{u}^\star)\mathrm{d}\boldsymbol{u}^\star\;\mathrm{d}R^\star, \\ Q^-(R,\boldsymbol{u}) &= -\nu^{bup}(R,\boldsymbol{u})f^R(R,\boldsymbol{u}),\end{aligned} \tag{1.86}$$

where $\Omega_v = \mathbb{R}^d_{\boldsymbol{u}} \times]R, +\infty[$, and ν^{bup} is the mean break-up frequency for a droplet with radius R and velocity $\boldsymbol{u}$. Besides, $h(R, \boldsymbol{u}, R^\star, \boldsymbol{u}^\star)$ is the number of droplets with radius R and velocity $\boldsymbol{u}$ produced by a $R^\star$ radius and $\boldsymbol{u}^\star$ velocity droplet. The term Q^+ models R radius and $\boldsymbol{u}$ velocity droplet formation from breakage of droplet with $R^\star$ radius and $\boldsymbol{u}^\star$ velocity. On the other hand, Q^- models disappearance of such droplets by breakage. The break-up frequency can be given by:

$$\nu^{bup}(R, \boldsymbol{u}) = \frac{||\boldsymbol{U_g} - \boldsymbol{u}||}{10R}\sqrt{\frac{\rho_g}{\rho_l}}. \tag{1.87}$$

In order to model fragment size and number, h is written:

$$h(R, \boldsymbol{u}, R^\star, \boldsymbol{u}^\star) = g_{\boldsymbol{u}}(R, \boldsymbol{u}, R^\star, \boldsymbol{u}^\star)\, g_R(R, R^\star, \boldsymbol{u}^\star), \tag{1.88}$$

where $\int g_{\boldsymbol{u}}(R, \boldsymbol{u}, R^\star, \boldsymbol{u}^\star)\mathrm{d}\boldsymbol{u} = 1$. Forms for g_R are proposed in (O'Rourke and Amsden 1987) and (Dufour 2005). They are closed using both the Sauter mean diameter of the resulting distribution:

$$R_{32}(R^\star, \boldsymbol{u}^\star) = \frac{\int_0^{+\infty} R^3 g_R(R, R^\star, \boldsymbol{u}^\star)\mathrm{d}R}{\int_0^{+\infty} R^2 g_R(R, R^\star, \boldsymbol{u}^\star)\mathrm{d}R}, \tag{1.89}$$

and mass conservation:

$$\int_0^{+\infty} R^3 g_R(R, R^\star, \boldsymbol{u}^\star)\mathrm{d}R = R^\star. \tag{1.90}$$

Finally, the Sauter mean diameter can be evaluated through correlations proposed in (Wert 1995):

$$\frac{R_{32}}{R^\star} = \begin{cases} 0.32(\mathrm{We}^\star)^{-1/3}\left(\dfrac{4.1}{(\mathrm{We}^\star - 12)^{1/4}}\right)^{2/3} & \text{if} \quad \mathrm{We}^\star \in]12, 18], \\ 0.32(\mathrm{We}^\star)^{-1/3}\left(\dfrac{2.45\sqrt{\mathrm{We}^\star - 12} - 1.9}{(\mathrm{We}^\star - 12)^{1/4}}\right)^{2/3} & \text{if} \quad \mathrm{We}^\star \in]18, 45], \\ 0.32(\mathrm{We}^\star)^{-1/3}\left(\dfrac{12.2}{(\mathrm{We}^\star - 12)^{1/4}}\right)^{2/3} & \text{if} \quad \mathrm{We}^\star \in]45, +\infty], \end{cases} \tag{1.91}$$

where $\mathrm{We}^\star$ is the Weber number associated to the $R^\star$ radius, $\boldsymbol{u}^\star$ velocity droplet. For $g_{\boldsymbol{u}}$, we take, see (Dufour 2005):

$$g_{\boldsymbol{u}}(R, \boldsymbol{u}, R^\star, \boldsymbol{u}^\star) = \delta(\boldsymbol{u} - \boldsymbol{u}^{bup}(R, R^\star, \boldsymbol{u}^\star)), \tag{1.92}$$

the mean droplet velocity after break-up $\boldsymbol{u}^{bup}$ can be given by Faeth correlation, see (Hsiang and Faeth 1993):

$$\boldsymbol{u}^{bup}(R, R^\star, \boldsymbol{u}^\star) = U_g + \frac{\boldsymbol{u}^\star - U_g}{1 + 2.7\left(\dfrac{\rho_g}{\rho_l}\dfrac{R^\star}{R}\right)^{2/3}}. \tag{1.93}$$

1.3.4 Dimensionless formulation

In order to obtain the liquid source terms for the dimensionless gas system, Eq. (1.25), we present the main elements in obtaining liquid dimensionless formulation, focusing on the $\boldsymbol{F}$, R_s and E coefficients. The liquid dimensionless formulation are driven by the gas one and thus we use the same reference quantities, see Sec. 1.1.2. We also introduce a reference number of droplets N_0, and a reference droplet density $n_0 = N_0/x_0^d$, for a physical space with d dimensions. Furthermore, for the droplet size description, we want to introduce a reference size, for example S_0 for the surface variable. In this case we introduce a reference surface that is not based on the gaseous reference length x_0. Two different reference times can then be defined:

$$\tau_{\text{drop}} = \frac{S_0}{x_0 U_0} \qquad \tau_g = \frac{x_0}{U_0}. \tag{1.94}$$

We define the dimensionless number $\mathrm{R}_{\text{drop-gas}}$:

$$\mathrm{R}_{\text{drop-gas}} = \frac{\tau_{\text{drop}}}{\tau_g} = \frac{S_0}{x_0^2}, \tag{1.95}$$

that will naturally arise in the dimensionless droplets terms, due to the cohabitation of these gas and liquid reference quantities. In this framework, the dimensionless number density function $\overline{f}$ can be written, for a d-space dimension domain:

$$\overline{f}\left(\overline{t}, \overline{x}, \overline{S}, \overline{\boldsymbol{u}}, \overline{T}\right) = f\left(t, x, S, \boldsymbol{u}, T\right) \frac{U_0^d \; S_0 \; T_\infty}{n_0}. \tag{1.96}$$

The dimensionless formulation for the coefficients $\boldsymbol{F}$, R_s, and E, is obtained as described below:

$$\begin{aligned} \overline{\boldsymbol{F}} &= \frac{x_0}{U_0^2}\,\boldsymbol{F}, \hat{\mathrm{A}} \\ \overline{R_s} &= \frac{x_0}{U_0\, S_0}\, R_s, \\ \overline{E} &= \frac{x_0}{U_0\, T_\infty}\, E. \end{aligned} \tag{1.97}$$

We finally obtain the following forms, where, for the sake of simplicity, we drop the $\overline{(.)}$ notation, all the variables being dimensionless:

$$\boldsymbol{F} = \begin{cases} \dfrac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p^\star}, & \mathrm{Re}_{\mathrm{p}}^\star < \mathrm{Re}_{\mathrm{p1}}^\star \\ \dfrac{(\boldsymbol{U_g} - \boldsymbol{u})}{\tau_p^\star}\left(1 + \dfrac{\mathrm{Re}_{\mathrm{p}}^{\star\,2/3}}{6}\right), & \mathrm{Re}_{\mathrm{p1}}^\star < \mathrm{Re}_{\mathrm{p}}^\star < \mathrm{Re}_{\mathrm{p2}}^\star \\ \dfrac{3\,\rho_g\,\sqrt{\pi}}{4\,\rho_l\, S^{1/2}}\, 0.45\; (\boldsymbol{U_g} - \boldsymbol{u})\, ||\boldsymbol{U_g} - \boldsymbol{u}||, & \mathrm{Re}_{\mathrm{p2}}^\star < \mathrm{Re}_{\mathrm{p}}^\star < \mathrm{Re}_{\mathrm{p3}}^\star \end{cases} \tag{1.98}$$

$$R_s = \frac{1}{\mathrm{R}_{\text{drop-gas}}}\, 4\pi\, \frac{\rho_g}{\rho_l}\, \mathrm{Sh_c}\, D_{Y_F}^\star \ln(1 + \mathrm{B_M}),$$

$$E = \frac{1}{\mathrm{R}_{\text{drop-gas}}}\, \frac{6\pi\,\rho_g}{\rho_l S\, C_{p,l}} \mathrm{Sh_c}\, D_{Y_F}^\star \ln(1 + \mathrm{B_M}) \left(\frac{C_{p,g}(T_g(x_p) - T)}{\mathrm{B_T}} - L_v\right),$$

where

$$\tau_p^\star = \frac{\rho_l\, S}{18\,\pi\,\mu_g^\star}, \tag{1.99}$$

and

$$\mathrm{Re}_\mathrm{p}^\star = \frac{\mathrm{Re}_\mathrm{p}\mathrm{Re}_0}{\mathrm{Re}_{\mathrm{p}0}} = \frac{\rho_g\; S^{1/2}\,\mathrm{R}_{\text{drop-gas}}^{1/2}\,||\boldsymbol{U_g}-\boldsymbol{u}||}{\sqrt{\pi}\,\mu_g^\star}, \tag{1.100}$$

with $\mathrm{Re}_{\mathrm{p}0}$ given by:

$$\mathrm{Re}_{\mathrm{p}0} = \frac{\rho_\infty {S_0}^{1/2} U_0}{\mu_\infty}, \tag{1.101}$$

and Re_0 by Eq. (1.20). The expressions for $\mu_g^\star$ and $D_{Y_F}^\star$ are given in Eq. (1.22). The values for $\mathrm{Re}_{\mathrm{p}_i}^\star$ are given by $\mathrm{Re}_{\mathrm{p}_i}^\star = \mathrm{Re}_{\mathrm{p}_i}\mathrm{Re}_{\mathrm{p}0}/\mathrm{Re}_0$, the values of $\mathrm{Re}_{\mathrm{p}_i}$ being obtained from Eq. (1.42). The dimensionless numbers Pr and Sc have to be computed with dimensionless values, see Eq. (1.23).

1.4 Approximation and simplified models

In order to study the relevance of a spray resolution method, the multi-fluid method in the present work, one needs to characterize precisely its behavior through comparisons with well-established spray methods. This characterization requires to develop well-controlled configurations where problems linked to the spray resolution choice are isolated. A one-way coupling with a simplified gas description considering low Mach number dilatable or incompressible flows is thus used in this framework. This approach allows to focus on the differences between various spray resolution methods coupled to the same gas description. Our interest being in combustion applications, we propose in this framework a first combustion model, allowing to focus on the impact of the spray resolution method.
Nevertheless, the spray resolution methods considered must be extendable to a more general framework. This is ensured in the present study by the high genericity used for discretization scheme derivation, Chpt. 3, and numerical tool design, Chpt. 6 and Chpt. 8.

1.4.1 One-way coupling

Taking two-way effects, i.e., influence of the spray on the gaseous carrier phase, may be of great importance for several application fields. Indeed if, for example, the spray is dense, it may modify significantly the mass and momentum of the gas. For combustion applications, the two-way coupling is necessary in order to take into account the vaporization of the droplets into the gaseous phase before burning. Nevertheless, we restrict ourselves in this study to one-way coupling. Indeed our objective in this work is to evaluate one level of description of spray, derived from the kinetic modeling, the Eulerian multi-fluid model. This evaluation aims at characterizing precisely the method, from its derivation to the numerical scheme used, from a given kinetic model. Two-way effects, as well as droplet models, are done at the kinetic level of description and the evaluation of their efficiency to reproduce

a complex physics is out of the scope of this study. Furthermore, analyzing the difficulties related to the spray resolution method itself is a necessary first step in the evaluation of the whole two-phase flow description.

This evaluation will be done through comparisons with a Lagrangian method derived from the same kinetic model. In this case, we want to take the Lagrangian method as a reference and taking into account two-way coupling will introduce a bias coming from the difficulties in coupling Eulerian and Lagrangian descriptions. Furthermore, the comparison of two-way effects modeling in Lagrangian and Eulerian descriptions can take advantage of the present study in order to focus only on coupling effects. Besides, we perform, in this study, comparisons between Eulerian multi-fluid method and experiments. To do so we use well-controlled experiments where the physics can be captured with simple modeling. Furthermore the experiments are done with a diluted, weakly evaporating to non-evaporating spray, so that two-way effects do not need to be modeled.

Finally, the important point we want to make is that this restriction is definitely not related with a limit of the multi-fluid model, where two way coupling can be accounted for, see (Massot, Kumar, Gomez, and Smooke 1998; Laurent 2002b; Laurent, Santoro, Noskov, Gomez, Smooke, and Massot 2004). Actually, it is, as for all Eulerian spray descriptions, far more easy than with a Lagrangian description. This restriction is purely due to the willingness to develop a simple framework for precise multi-fluid model characterization.

1.4.2 Incompressible and dilatable low Mach flows

Low Mach number formulation

We consider in this work gas flow under the low Mach number assumption. In this case the spatial pressure changes can be neglected. The pressure can thus be written: $P_g(\boldsymbol{x},t) = P_{\rm atm}(t) + \tilde{P}(\boldsymbol{x},t)$, where $\tilde{P}$ is a small perturbation: $\frac{\tilde{P}}{P_{\rm atm}} = O(M^2) << 1$, M being the Mach number. Under this assumption, the power produced by the viscous stress can be neglected, as $\boldsymbol{\tau} : \partial_{\boldsymbol{x}} \cdot (\boldsymbol{U_g}) = O(M^2)$, see for example (Giovangigli 1999a) and references therein. Finally, the dimensionless gas equation system, Eq. (1.25), becomes, for the chosen one-way modeling:

$$\begin{aligned}
\partial_t \rho_g + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g}) &= 0, \\
\partial_t(\rho_g Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g (\boldsymbol{U_g} + \boldsymbol{U_g^{\rm cor}}) Y_k) &= \partial_{\boldsymbol{x}} \cdot (\rho_g D_k^\star \partial_{\boldsymbol{x}} \cdot (Y_k)) + \dot{\omega}_k, \\
\partial_t(\rho_g \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} \otimes \boldsymbol{U_g}) &= -\frac{1}{\gamma_\infty M^2} \partial_{\boldsymbol{x}}(\tilde{P}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}), \\
\partial_t(\rho_g h_s) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g}\, h_s) &= \frac{\gamma_\infty - 1}{\gamma_\infty} \partial_t(P_g) - \partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}^\star) + \dot{\omega}_T.
\end{aligned} \tag{1.102}$$

Dilatable and incompressible flows

In this work, we study an Eulerian multi-fluid method for spray, that is mainly characterized in configuration without combustion. Furthermore we focus on the evaluation of the

numerical method for the dispersed liquid phase and thus the gas flow description is not assessed within this study. We therefore reduce to low Mach number dilatable flows where we do not solve the mass conservation equation Eq. (1.1), and where the density of gas is obtained with the state law for perfect gases:

$$\rho_g = \frac{P_{\text{atm}}\, w_g}{R\, T_g}, \tag{1.103}$$

where w_g is the mixture molar mass:

$$w_g = \left(\sum_{k=1}^{n_s} \frac{Y_k}{w_k} \right)^{-1}, \tag{1.104}$$

and w_k the k^{th} species molar mass. The system of gaseous equation reduces in this case to:

$$\begin{aligned}
\rho_g &= \frac{P_{\text{atm}}\, w_g}{R\, T_g}, \\
\partial_t \rho_g + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g}) &= 0, \\
\partial_t(\rho_g Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g (\boldsymbol{U_g} + \boldsymbol{U_g^{\text{cor}}}) Y_k) &= \partial_{\boldsymbol{x}} \cdot (\rho_g D_k^\star \partial_{\boldsymbol{x}} \cdot (Y_k)) + \dot{\omega}_k, \\
\partial_t(\rho_g \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} \otimes \boldsymbol{U_g}) &= -\frac{1}{\gamma_\infty M^2} \partial_{\boldsymbol{x}}(\tilde{P}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}), \\
\partial_t(\rho_g h_s) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g}\, h_s) &= \frac{\gamma_\infty - 1}{\gamma_\infty} \partial_t(P_g) - \partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}^\star) + \dot{\omega}_T.
\end{aligned} \tag{1.105}$$

Finally, in order to obtain a simple gas modeling to study easily the response of different spray description, we reduce in some cases to incompressible flow assumption. In this zero-divergence fluid, the system of equations can be written:

$$\begin{aligned}
\rho_g &= \frac{P_{\text{atm}}\, w_g}{R\, T_g}, \\
\partial_{\boldsymbol{x}} \cdot (\boldsymbol{U_g}) &= 0, \\
\partial_t(\rho_g Y_k) + \partial_{\boldsymbol{x}} \cdot (\rho_g (\boldsymbol{U_g} + \boldsymbol{U_g^{\text{cor}}}) Y_k) &= \partial_{\boldsymbol{x}} \cdot (\rho_g D_k^\star \partial_{\boldsymbol{x}} \cdot (Y_k)) + \dot{\omega}_k, \\
\partial_t(\rho_g \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} \otimes \boldsymbol{U_g}) &= -\frac{1}{\gamma_\infty M^2} \partial_{\boldsymbol{x}}(\tilde{P}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau}), \\
\partial_t(\rho_g h_s) + \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g}\, h_s) &= \frac{\gamma_\infty - 1}{\gamma_\infty} \partial_t(P_g) - \partial_{\boldsymbol{x}} \cdot (\boldsymbol{q}^\star) + \dot{\omega}_T.
\end{aligned} \tag{1.106}$$

1.4.3 A first model for two-phase flow combustion

In the framework of developing a simplified gaseous description to focus on the study of the spray carried, we only take into account one species for the fuel, with mass fraction Y_F,

and one for the oxidizer, with mass fraction Y_{Ox}. An important feature of spray models evaluation, when aiming at combustion applications, is to assess their ability to simulate accurately gaseous fuel production through the evaporation process. In order to assess this key issue, still using a framework devoted to spray resolution evaluation, the one-way assumption is slightly modified in order to obtain a first idea of fuel mass fraction prediction by the studied spray models. Furthermore, we also develop a simplified model to take into account fuel mass fraction consumption by combustion. The computations using the models described below are presented in Chpt. 10.

Gaseous fuel mass fraction prediction

In order to study gaseous fuel production by the spray evaporation without modifying the gas phase, we define $Y_{F'}$ as a passive scalar, advected by the flow, where we store the gaseous fuel produced by the spray evaporation. The gas phase composition is not modified by the spray evaporation, so that we stay in a one-way coupling assumption. Nevertheless we can assess the evaporation source term given by the spray model, and feeding the passive scalar $Y_{F'}$. This scalar is advected and diffused by the flow, and fed by the spray evaporation:

$$\partial_t\left(\rho_g\, Y_{F'}\right) + \partial_{\boldsymbol{x}}\cdot\left(\rho_g\left(\boldsymbol{U_g}+\boldsymbol{U_g}^{\rm cor}\right)Y_{F'}\right) = \partial_{\boldsymbol{x}}\cdot\left(\rho_g\, D^{\star}_{Y_{F'}}\partial_{\boldsymbol{x}}\cdot\left(Y_{F'}\right)\right) + S_F^{species}. \tag{1.107}$$

This procedure has two advantages. First, it allows to have an insight of the fuel mass fraction evolution, without dealing with two-way coupling difficulties. Second, as this "virtual" evaporation does not modify the gas phase dynamics, it is possible to compare the evaporation produced by two different spray methods with the same gas phase simulation, allowing to derive precise comparisons focusing only on the spray methods aspects. We will refer in the following to this procedure as being the FADO procedure for Fuel mAss Description with One-way coupling.

Two Phase Coldflame

We are also interested in taking combustion into account in the equation of $Y_{F'}$:

$$\partial_t\left(\rho_g\, Y_{F'}\right) + \partial_{\boldsymbol{x}}\cdot\left(\rho_g\left(\boldsymbol{U_g}+\boldsymbol{U_g}^{\rm cor}\right)Y_{F'}\right) = \partial_{\boldsymbol{x}}\cdot\left(\rho_g\, D^{\star}_{Y_{F'}}\partial_{\boldsymbol{x}}\cdot\left(Y_{F'}\right)\right) + \dot{\omega}_{Y_{F'}} + S_F^{species}. \tag{1.108}$$

To do so, we will use a simple model with a global one-step non-reversible reaction

$$\nu_{F'}\, Y_{F'} + \nu_{Ox'}\, Y_{Ox'} \rightarrow \nu_{Pr'}\, Y_{Pr'}, \tag{1.109}$$

where $Y_{F'}$, $Y_{Ox'}$, and $Y_{Pr'}$ are the mass fraction of the fuel, the oxidizer and the products, respectively. $\nu_{F'}$, $\nu_{Ox'}$ and $\nu_{Pr'}$ are their stoichiometric coefficients. The ratio $\nu_{Ox'}/\nu_{F'}$ is set to 15, so that it corresponds to the stoichiometry of the combustion of n-heptane with air. The important point that one has to notice is that we want to develop this simple combustion model in a one-way coupling framework. Therefore, $Y_{F'}$, $Y_{Ox'}$ and $Y_{Pr'}$ are not taking part in the gaseous phase composition. They can be seen as passive scalar with formation or destruction source terms, not taken into account in the thermodynamics computation. The evolution of $Y_{F'}$, Eq. (1.108), has thus to be coupled with the evolutions of $Y_{Ox'}$ and $Y_{Pr'}$, given by:

$$\begin{aligned}
\partial_t\left(\rho_g\, Y_{Ox'}\right) + \partial_{\boldsymbol{x}}\cdot\left(\rho_g\left(\boldsymbol{U_g}+\boldsymbol{U_g}^{\rm cor}\right)Y_{Ox'}\right) &= \partial_{\boldsymbol{x}}\cdot\left(\rho_g\, D^{\star}_{Y_{Ox'}}\partial_{\boldsymbol{x}}\cdot\left(Y_{Ox'}\right)\right) + \dot{\omega}_{Y_{Ox'}},\\
\partial_t\left(\rho_g\, Y_{Pr'}\right) + \partial_{\boldsymbol{x}}\cdot\left(\rho_g\left(\boldsymbol{U_g}+\boldsymbol{U_g}^{\rm cor}\right)Y_{Pr'}\right) &= \partial_{\boldsymbol{x}}\cdot\left(\rho_g\, D^{\star}_{Y_{Pr'}}\partial_{\boldsymbol{x}}\cdot\left(Y_{Pr'}\right)\right) + \dot{\omega}_{Y_{Pr'}}.
\end{aligned}$$

(1.110)

Then we do not compute any heat release or gas dilatation, we only consider a "virtual" fuel and oxidizer consumption and "virtual" products formation. The gas phase is not modified by this combustion, that we will call Coldflame combustion.

The formation or destruction rate for the "virtual" species are obtained from the reaction rate $\dot{\omega}_r$:

$$\begin{aligned} \dot{\omega}_{Y_{F'}} &= -\nu_{F'}\,\dot{\omega}_r, \\ \dot{\omega}_{Y_{Ox'}} &= -\nu_{Ox'}\,\dot{\omega}_r, \\ \dot{\omega}_{Y_{Pr'}} &= (\nu_{F'}+\nu_{Ox'})\,\dot{\omega}_r. \end{aligned} \tag{1.111}$$

The reaction rate, for a classical Arrhenius law, is given by:

$$\dot{\omega}_r = K\,\rho_g\,Y_{F'}\,Y_{Ox'}\,\exp\left(\frac{-T_a}{T_g}\right), \tag{1.112}$$

where K is a constant, and T_a is the activation temperature, (Poinsot and Veynante 2005). We introduce the temperature factor, α, and the reduce activation energy, β, see (Poinsot and Veynante 2005), given by:

$$\alpha = \frac{T_b - T_\infty}{T_b}; \quad \beta = \alpha\frac{T_a}{T_b}, \tag{1.113}$$

where T_b and T_∞ are the temperatures of burnt and fresh gases, respectively. We choose classical values for these parameters: $\alpha = 0.8$ and $\beta = 8$, based on standard heat release and flame thickness, see examples in (Poinsot and Veynante 2005). It assumes a burnt gases temperature T_b five times greater than fresh gases temperature T_∞. With these definition, the reaction rate becomes:

$$\dot{\omega}_r = K\,\rho_g\,Y_{F'}\,Y_{Ox'}\,\exp\left(\frac{\beta}{\alpha} - \frac{\beta}{\alpha(1-\alpha)}\frac{T_\infty}{T}\right), \tag{1.114}$$

where the term $\exp(-\beta/\alpha)$ has been introduced in the constant K. Furthermore, in order to correct the classical flame speed error introduced by one-step chemistry for equivalence ratio $\phi = (Y_{F'}/Y_{Ox'})/\,(Y_{F'}/Y_{Ox'})_{stoich} > 1$, we used a GKAS technique, (Vervisch, Labegorre, and Reveillon 2004; Canneviere 2003), giving a maximum flame speed at $\phi = 1$. It yields the reaction rate:

$$\dot{\omega}_r = K(\phi)\,\rho_g\,Y_{F'}\,Y_{Ox'}\,\frac{s(Y_{F'}+Y_{Ox'})^2}{(sY_{F'}+Y_{Ox'})^2}\exp\left(\frac{\beta}{\alpha} - \frac{\beta}{\alpha(1-\alpha)}\frac{T_\infty}{T}\right), \tag{1.115}$$

where $s = \nu_{Ox'}/\nu_{F'}$. Finally, as the temperature is not known in the Coldflame computation, we extrapolate it from the burnt gases mass fraction:

$$T = T_\infty + (T_b - T_\infty)\,Y_{Pr'}, \tag{1.116}$$

with $T_b = 5\,T_\infty$, as assumed for the temperature factor α. It gives in dimensionless framework:

$$T = 1 + 4\,Y_{Pr'}. \tag{1.117}$$

As far as ignition is concerned, we set the chemical equilibrium in the whole flow at a fixed time, the initial time for combustion, and afterward we let the combustion evolve. We thus add burnt gases and withdraw the fuel and oxidizer that would have lead to these products. To do so, we assume an infinitely fast chemistry and we use the Burke-Schumann flame structure, see (Poinsot and Veynante 2005). The mass fractions thus depends on the relation between the mixture fraction, Z and its stoichiometric value $(Z)_{stoich}$, defined by:

$$Z = \frac{\nu_{Ox'} Y_{F'} - \nu_{F'} Y_{Ox'} + 1}{\nu_{Ox'} + \nu_{F'}}, \quad (Z)_{stoich} = \frac{1}{\nu_{Ox'} + \nu_{F'}}. \tag{1.118}$$

We thus have the following structure:

- in the rich zone, $Z > (Z)_{stoich}$:

$$\begin{aligned} Y_{Ox'} &= 0, \\ Y_{F'} &= \frac{Z - (Z)_{stoich}}{1 - (Z)_{stoich}}, \\ Y_{Pr'} &= 1 - Y_{F'}, \end{aligned} \tag{1.119}$$

- in the lean zone, $Z < (Z)_{stoich}$:

$$\begin{aligned} Y_{F'} &= 0, \\ Y_{Pr'} &= \frac{Z}{(Z)_{stoich}}, \\ Y_{Ox'} &= 1 - Y_{Pr'}. \end{aligned} \tag{1.120}$$

This model will allow us to study a first combustion case with the spray method developed and characterized within this work. Furthermore, it will also enable us to compare the behavior of different spray resolution methods with combustion, in a one-way framework, thus isolating the questions related to the spray resolution itself.

Conclusion

The background assumptions of the spray kinetic description, as well as the physical models chosen to describe spray evolution, have been provided. This kinetic framework cannot be used at the scale of realistic configurations, given the number of dimensions of the associated phase space. Nevertheless it provides the basis for macroscopic model derivations, described in Chpt. 2.
Furthermore, this chapter provides a simplified framework devoted to spray resolution evaluation, that is used to conduct precise characterization of the method studied in this work, Chpt. 4, Chpt. 9, Chpt. 11. This framework allows to assess combustion issues, Chpt. 10.

Chapter 2

Lagrangian and Eulerian resolution methods for kinetic spray description

Contents

This chapter discusses the choice of the resolution method for the spray among different available strategies. We introduced in Chpt. 1, the framework of spray modeling and we focus here on its simulation. Direct resolution of the kinetic equation Eq. (1.30) can difficulty be considered for realistic configurations, given the high number of phase space dimensions in a 3-D case. Alternative methods have thus to be considered. A natural choice for spray resolution method is a Lagrangian method where we track particles in the flow, considered as representing physical droplets or as statistical particles according to the type of approach. This widely used approach, has proven its great efficiency to simulate complete spray dynamics without introducing any numerical diffusion, (Dukowicz 1980; O'Rourke 1981; Amsden, O'Rourke, and Butler 1989; Reveillon, Bray, and Vervish 1998; Miller and Bellan 1999; Rüger, Hohmann, Sommerfeld, and Kohnen 2000). Eulerian spray description represents, however, an interesting complementary tool, given its easier coupling with the gas phase description, and its easier optimization through parallel computing, making it an interesting choice for High Performance Computing (HPC), (Boileau 2007; Lamarque 2007). We summarize here the main Lagrangian approaches, highlighting their level of modeling in order to study their connection with the kinetic level of description, Sec. 1.2, and thus to establish their link with the Eulerian approaches. The connection between Lagrangian and Eulerian approaches is a key issue when conducting comparisons between the two, as done in Chpt. 4 and throughout Part. IV. We then present the main derivation strategies leading to Eulerian moment approaches, along with the associated assumptions. Among this methods, we focus on the multi-fluid model, (Greenberg, Silverman, and Tambour 1993; Laurent and Massot 2001), a moment methods for spray describing polydispersion, a key issue for two-phase flows combustion applications. The spray computations presented in this work, Part. II and Part. IV, are done in the context of a Direct Numerical Simulation, and thus coupled with a DNS of the gas phase. This framework, allows to focus on the spray resolution methods, and to provide precise characterization of the chosen methods, as done in Part. II. Furthermore, when studying the multi-fluid model, it provides a necessary first step toward LES formulation. We insist that, albeit this approach needs further modeling, the described multi-fluid resolution method, along with its numerical schemes, given in Chpt. 3, still applies in the context of LES.

2.1 Lagrangian particle discretization

We propose in this section to first go through the main approaches of Lagrangian particular descriptions of spray, highlighting the associated level of description. We then describe the Direct Simulation Monte-Carlo (DSMC) method, that is the Lagrangian description derived from the kinetic level of description presented in Sec. 1.2. Finally, in order to derive a Lagrangian method with a reasonable cost for unsteady polydisperse flows, we consider one realization of the statistical Lagrangian cloud, and highlight its link with Discrete Particle

Simulation (DPS) method in the infinite Knudsen limit, i.e., for non-collisional spray, as viewed in this study. We study here DNS framework, all the scales of the gas flow are thus resolved. When considering modeled turbulent carrier phase, extra effort has to be done to describe particle turbulent dispersion. We do not tackle this issue in this work as we focus on DNS, and we refer to (Crowe, Sommerfeld, and Tsuji 1998) for a review of the turbulent dispersion modeling approaches. Apart from this extra modeling term, all the presented issues stand in modeled or partially computed gas turbulent flows.

2.1.1 Lagrangian techniques for spray description

We present in the following three Lagrangian techniques for spray resolution, based on deterministic or on statistic spray description.

Discrete Particle Simulation

The first Lagrangian method associated to the mesoscopic dispersed-phase description is the Discrete Particle Simulation (DPS). In this framework, each Lagrangian numerical particles represent a physical droplets. A system of ODE is thus solved to obtain the evolution of the droplet parameters, ϕ_k, $\boldsymbol{u}_k$ and T_k for the k^{th} droplet. This two-phase DNS has been introduced in (Riley and Patterson 1974). Most of the first numerical studies were dedicated to solid particle dispersion, (Squires and Eaton 1991; Elghobashi and Truesdell 1992). The extension to evaporating droplets in turbulent flows has been provided in (Mashayek, Jaberi, Miller, and Givi 1997; Reveillon, Bray, and Vervish 1998; Miller and Bellan 1999), and has been used in combustion applications in (Mashayek 1998; Reveillon and Vervisch 2005; Miller and Bellan 2000). This method, is mainly used in DNS configuration, where all the droplets contained in the physical domain can be tracked.

Statistical Lagrangian discretization

The statistical Lagrangian description can be used in two different contexts.
First it can be used instead of a DPS method in a configuration where the high number of physical droplets prevents to use one numerical particle for each droplet, given the high computational cost associated. In this framework, the numerical particle, also called parcel, represent several physical droplets associated. This description has been introduced for fuel spray in (O'Rourke 1981), extending the work of (Dukowicz 1980). This method is referred in (Crowe, Sommerfeld, and Tsuji 1998) as the discrete element method, and as a multicontinua method in (Sirignano 1999), where the parcels are defined as classes of droplets. The computational cost of this method is obviously linked to the chosen number of tracked parcels. The cost of the method is thus well controlled, and it is therefore broadly used to compute industrial configurations, This method is implemented for instance in KIVA code, (Amsden, O'Rourke, and Butler 1989), and is used in many industrial computational codes at present. We refer to this method in this work as Stochastic Parcel (SP) method, as done in (O'Rourke 1981).
On the other hand, the statistical Lagrangian method can be seen as a resolution method of the Williams kinetic equation. This approach is called Direct Simulation Monte-Carlo

method (DSMC), and is originally introduced in (Bird 1994) for rarefied gas. This approach aims at describing the evolution of the spray NDF moments, and thus a high number of statistical particles are needed to obtain a converged solution. In this case, several numerical particles may be needed to represent one physical droplet. The numerical particle has a weight associated, adapted to the needed refinement. This method provides directly the ensemble average, in terms of initial condition, and then the reconstructed Eulerian fields correspond to the same level of information than the one provided by a Eulerian method. The difference between the two methods, is only the level of refinements provided. Indeed, the convergence expected in the DSMC method demands a high number of statistical particles, leading to a higher refinement level than in DPS. On the contrary, the SP method aims at coarsening the DPS description. The number of statistical parcels is thus set by the computational cost limitation, without drawing any links with kinetic level of description.

In this study, we use different types of Lagrangian methods, depending on the issues tackled. In order to conduct Eulerian-Lagrangian fine comparisons to characterize the multi-fluid model, as done in Chpt. 4, we use the DSMC method, as it provides the same level of description than a kinetic-based Eulerian method. Indeed we are comparing two ways of discretizing the Williams equation, Eq. (1.30). Nevertheless, when assessing the cost of the Eulerian methods, Chpt. 9, it should be compared to a Lagrangian technique closer to industrial concern. A SP method could be used, nevertheless, as in the DNS configurations studied all the physical droplets can be tracked, a DPS technique is employed, see Chpt. 9, Chpt. 10 and Chpt. 11. We recall here that the DPS deterministic tracking is coupled with a statistial initialization, as the initial conditions can not be known in a deterministic way. Therefore this description can be seen as one realization of the statistical DSMC description, and thus contains less information.

2.1.2 Direct Simulation Monte-Carlo method

Statistical formulation

The DSMC method can be seen as a stochastic resolution method for the Williams equation, Eq. (1.30) of the kinetic formulation, see for example (Hylkema 1999) and references therein. The spray NDF is written as a sum of Dirac delta functions:

$$f(t,\boldsymbol{x},\phi,\boldsymbol{u},T)=\sum_{k=1}^{N_p} w_k\,\delta(\boldsymbol{x}-\boldsymbol{x}_k(t))\,\delta(\phi-\phi_k(t))\,\delta(\boldsymbol{u}-\boldsymbol{u}_k(t))\,\delta(T-T_k(t)), \tag{2.1}$$

where w_k is the weight of the k^{th} numerical particle and $\boldsymbol{x}_k$, $\boldsymbol{u}_k$, ϕ_k, T_k are its position, velocity, size and temperature. These characteristics of numerical particles evolve through the equations:

$$\mathrm{d}_t\,\boldsymbol{x}_k=\boldsymbol{u}_k,\quad \mathrm{d}_t\,\boldsymbol{u}_k=\boldsymbol{F}_k,\quad \mathrm{d}_t\phi_k=-R_{\phi_k},\quad \mathrm{d}_t\,T_k=E_k, \tag{2.2}$$

where the expressions for $\boldsymbol{F}$, R_ϕ and E are given in Sec. 1.2. The dimensionless formulation used is already defined in Sec. 1.3.4, where the dimensionless variables are introduced, and the dimensionless droplet models terms expressions explicitly given.

Initialization

A specific attention must be devoted to the initialization of such a method, since it defines the number of numerical particles and has to be a sufficiently accurate realization of the initial distribution function. Two methods can be used to compute a realization of a given initial NDF f_0^ϕ:

- simulating a uniform distribution on the support of f_0^ϕ and adjusting the weights to obtain the correct distribution ;
- simulating directly the mass distribution, $\rho_l S^{3/2}/\left(6\sqrt{\pi}\right) f_0^\phi$, through a rejection method with numerical particles having weights representing the same mass density.

The first method, albeit easier to implement, induces particles with a very small mass density and thus little efficiency in terms of computational cost. Furthermore, this method, based on a uniform distribution, describes with the same precision the whole distribution, whereas the second, leading to a mass of droplets linked with the number density function form, gives a better precision for main part of the size phase space, while it coarsen the resolution for the tails of the distribution where few droplets are likely to be found. This second method gives thus a better global precision. We thus decided to use the rejection method to initialize the DSMC method. This method is based on Theorem 1.

Theorem 1 (Rejection method)**.** *Let p and q be two probability densities taking values in $\mathbb{R}^d$. Let us assume that there exists $C \geq 1$ such that:*

$$\forall x \in \mathbb{R}^d \quad Cq(x) \geq p(x). \tag{2.3}$$

Let $(Y_n)_{n\geq 1}$ be a sequence of independent and identically distributed random vectors with density q.
Let $(U_n)_{n\geq 1}$ be a sequence of independent and identically distributed random vectors uniformly distributed on $[0,1]$.
Then $X = Y_U$, with $U = \inf\{k \geq 1, CU_k q(Y_k) < p(Y_k)\}$, is a random vector with density p.

In order to illustrate the application of this theorem to the initialization of the spray NDF, we assume here that the droplets have no initial velocity and that the droplet spectrum of the spray is independent of the space location at the initial time, and thus we only consider the size phase space and physical space initialization, $f_0(\boldsymbol{x}_k, \phi_k) = \mathbf{1}_\Omega(\boldsymbol{x}_k)\, f_0^\phi(\phi_k)$. One has to note that this restriction is only for illustration purpose and that the rejection method can handle any form of the initial NDF. We use the rejection method to set numerical parcels with the distribution f_0, taken as the distribution p of Theorem 1. A change of variables is possibly performed so that the support of this initial NDF f_0 becomes $\Omega \times [0,1]$, where $\Omega \subset [0,1]\times[0,1]\times[0,1]$ is the space where droplets can be found. The global initial number distribution f_0 is bounded by a constant C, taken as $\max_{x,\phi}(f_0)$. Initial conditions can be obtained in the following way : five random numbers $(X_k, Y_k, Z_k, \Phi_k, U_k)$ between 0 and 1 are chosen; if $\boldsymbol{x}_k = (X_k, Y_k, Z_k)$ belongs to Ω and if $f_0(\boldsymbol{x}_k, \Phi_k) > CU_k$, then the particle is accepted; if not, it is rejected. This is done till the number N_p of accepted numerical particles we want is reached. Then, each numerical particle (X_k, Y_k, Φ_k) has the weight $w_k = m_0^{\text{tot}}/\left(N_p\, Vol(\phi_k)\rho_l\right)$, where m_0^{tot} is the total initial mass density of the spray, and $Vol(\phi_k)$ the expression of the droplet volume in function of the size variable ϕ_k. For instance if the surface, S_k, is taken as the size variable: $Vol(S_k) = S_k^{3/2}/(6\sqrt{\pi})$.

The algorithm of this method is explicitly written in Chpt. 8. This method can thus be implemented through Algo. 1.

Algorithm 1 NDF initialization through rejection method

```
n := 0
while n<N_p do
  generate uniform random variables (X_k, Y_k, Z_k, Φ_k, U_k)
  x_k := (X_k, Y_k, Z_k)
  while x_k ∉ Ω or f_0^φ(φ_k) ≤ C U_k do
    generate uniform random variables (X_k, Y_k, Z_k, Φ_k, U_k)
    x_k := (X_k, Y_k, Z_k)
  end while
  x_n := x_k
  φ_n := φ_k
  n := n+1
end while
```

Eulerian ensemble average reconstruction

As previously mentioned, this statistical method gives access directly to the ensemble averages. The ensemble average on a considered Eulerian grid cell will then give access to the quantities assessed in the Eulerian computation and thus allow precise comparisons. Therefore we can define a space cell $\mathcal{C}_\alpha$ and reconstruct the Eulerian mass density and velocities from the Lagrangian simulation:

$$\begin{aligned} m_\alpha &= \int_{\mathcal{C}_\alpha}\int_\phi\int_{\boldsymbol{u}} \rho_l\, Vol(\phi)\, f \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\phi\,\mathrm{d}\boldsymbol{u} &&= \textstyle\sum_{\boldsymbol{x}_k\in\mathcal{C}_\alpha} w_k\,\rho_l\, Vol(\phi_k), \\ \boldsymbol{u}_\alpha &= \frac{1}{m_\alpha}\int_{\mathcal{C}_\alpha}\int_\phi\int_{\boldsymbol{u}} \rho_l\, Vol(\phi)\boldsymbol{u}\; f \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\phi\,\mathrm{d}\boldsymbol{u} &&= \textstyle\sum_{\boldsymbol{x}_k\in\mathcal{C}_\alpha} w_k\,\rho_l\, Vol(\phi_k)\,\boldsymbol{u}_k, \end{aligned} \tag{2.4}$$

where the summation is done over all the k such that $\boldsymbol{x}_k \in \mathcal{C}_\alpha$. In order to realize comparisons with an Eulerian method resolving the spray polydispersion, as the multi-fluid model, we can condition the average on the size of the droplets and define for example a size interval $[\phi_p, \phi_{p+1}]$, where the eulerian quantities become:

$$\begin{aligned} m_\alpha^{(p)} &= \int_{\mathcal{C}_\alpha}\int_{\phi_p}^{\phi_{p+1}}\int_{\boldsymbol{u}} \rho_l\, Vol(\phi)\, f \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\phi\,\mathrm{d}\boldsymbol{u} &&= \textstyle\sum \rho_l\, w_k\, Vol(\phi_k), \\ \boldsymbol{u}_\alpha^{(p)} &= \frac{1}{m_\alpha}\int_{\mathcal{C}_\alpha}\int_{\phi_p}^{\phi_{p+1}}\int_{\boldsymbol{u}} \rho_l\, Vol(\phi)\boldsymbol{u}\; f \,\mathrm{d}\boldsymbol{x}\,\mathrm{d}\phi\,\mathrm{d}\boldsymbol{u} &&= \textstyle\sum \rho_l\, w_k\, Vol(\phi_k)\,\boldsymbol{u}_k, \end{aligned} \tag{2.5}$$

where the summation is done over all the k such that $\boldsymbol{x}_k \in \mathcal{C}_\alpha$ and that $\phi_p < \phi_k < \phi_{p+1}$.

This method is thus used in Chpt. 4, to conduct precise quantitative comparisons between the Eulerian method of interest, the multi-fluid, and a Lagrangian approach. Nevertheless, the drawback of this method is the high number of numerical particles needed to obtain a precise description of the statistics in unsteady polydisperse configurations.

2.1.3 Discrete Particle Simulation

DPS requirement and link with Eulerian description

As we aim at showing in this study the feasibility of spray simulation with multi-fluid description in more realistic 2-D to 3-D applications. We also aim at providing an idea of the associated cost compared to a Lagrangian approach. We thus aim at using in this case a Lagrangian method closer to industrial concern than DSMC method. The widely used DPS method introduced in Sec. 2.1.1, is a good choice for such a purpose. The remaining question is how to compare it to kinetic-based Eulerian simulation, as the DPS is not based on a kinetic approach. This question can be answered in the infinite Knudsen limit, i.e., non-colliding, cases considered in the present study. Indeed, in this framework, DPS can be seen as one realization of a statistical Lagrangian description driven by the stoichiometry. We will thus use this method to assess the ability of the Eulerian method studied to reproduce the global physics of considered configurations, realizing qualitative comparisons between the Eulerian method and Lagrangian DPS. These comparisons will also allow to assess the efficiency of the Eulerian description of spray polydispersion.

Lagrangian DPS mass source term

This Lagrangian approach will be used to estimate the Eulerian method behavior for fuel gaseous mass fraction prediction and coldflame computation, Sec. 1.4.3. We give here the expression of the Lagrangian evaluation of the mass source term in the gaseous equations, $S_F^{species}$. In this DPS context, with mono-component droplets we can reconstruct this Eulerian field on the space cell $\mathcal{C}_\alpha$:

$$S_F^{species} = S^m = \sum_{\boldsymbol{x}_k \in \mathcal{C}_\alpha} \frac{\rho_l}{V_{\mathcal{C}_\alpha}} \, \mathrm{d}_t Vol(\phi_k(t)), \tag{2.6}$$

where $V_{\mathcal{C}_\alpha}$ is the volume of $\mathcal{C}_\alpha$, and where we sum over all the k such that $\boldsymbol{x}_k \in \mathcal{C}_\alpha$. We give here the expression of this term when choosing the droplet surface S_k as the size variable, since this size will be used for the multi-fluid formulation, Sec. 2.3. In this context we have:

$$Vol(S_k) = \frac{S_k^{3/2}}{6\sqrt{\pi}}, \quad \mathrm{d}_t Vol(S_k(t)) = \frac{S_k^{1/2}}{4\sqrt{\pi}} \, \mathrm{d}_t S_k = \frac{S_k^{1/2}}{4\sqrt{\pi}} \, R_{S_k}, \tag{2.7}$$

and therefore:

$$S_F^{species} = S^m = \sum_{\boldsymbol{x}_k \in \mathcal{C}_\alpha} \frac{\rho_l}{V_{\mathcal{C}_\alpha}} \frac{S_k^{1/2}}{4\sqrt{\pi}} \, R_{S_k}. \tag{2.8}$$

In the dimensionless setting previously introduced, Sec. 1.3.4, we have:

$$\overline{S_F^{species}} = \frac{x_0}{\rho_\infty U_0} \, S_F^{species}, \tag{2.9}$$

and thus, dropping the notation $\overline{(.)}$, all the variables being dimensionless:

$$S_F^{species} = S^m = \frac{\rho_l \, \mathrm{R_{drop\text{-}gas}}^{3/2}}{V_{\mathcal{C}_\alpha} \, 4\sqrt{\pi}} \sum_{\boldsymbol{x}_k \in \mathcal{C}_\alpha} S_k^{1/2} \, R_{S_k}, \tag{2.10}$$

where $\mathrm{R_{drop\text{-}gas}}$ is described in Sec. 1.3.4. Computing this term on the Eulerian grid leads to important difficulties, common to all the Lagrangian approaches, that are raised in Chpt. 8.

2.2 Eulerian moment methods for spray: strategy overview

The alternative to Lagrangian particle tracking is the resolution of spray Eulerian global quantities, as number or mass density, momentum and enthalpy. These Eulerian methods can be seen as moment methods derived from the kinetic equation Eq. (1.30). We propose to present here the main derivation strategies and resulting methods in the framework of spray. We highlight in this presentation the key issues of Eulerian spray modeling in combustion context, that are velocity and size distribution description. Indeed these two issues condition the ability of Eulerian methods to describe polydispersion and out-of-equilibrium velocity distribution, i.e, droplet trajectory crossings description. The derivation strategies are presented in a laminar framework, in order to highlight the key points of the moment equation derivation, without introducing extra modeling complexity. We then discuss the extension to turbulent flows and explain the choice of the Eulerian method studied in this work, a mono-kinetic multi-fluid method. Finally we detail the derivation of this very method.

2.2.1 Derivation strategies for DNS of kinetic-based spray

The full resolution of the kinetic equation, Eq. (1.30), named by O'Rourke the full spray equation method (O'Rourke 1981), can difficultly be used given its cost related to the high number of phase space dimensions. Indeed in a 3-D case, the phase space is of dimension height. Nevertheless, in many cases the knowledge of the full kinetic description of the spray is not needed, and it is sufficient to know the evolution of global quantities, the moments of the NDF moments. For a given function $\psi(y)$, the k^{th} order moment M_k is defined by:

$$M_k = \int_y y^k \, \psi(y) \, \mathrm{d}y. \tag{2.11}$$

Therefore, for the NDF we introduce the moment $M_{k\,\boldsymbol{l}\,m}$:

$$M_{k\,\boldsymbol{l}\,m} = \int_\phi \int_{\boldsymbol{u}} \int_T \phi^k \, \boldsymbol{u}^{\boldsymbol{l}} \, T^m \, f^\phi(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) \, \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \tag{2.12}$$

of order k in size, $|\boldsymbol{l}|$ in velocity and m in temperature. The variable, $\boldsymbol{l}$ defines a multi-index, needed for the different velocity components. We denote by $|\boldsymbol{l}|$ the order of the multi-index. Thus, for a first order velocity moment $|\boldsymbol{l}| = 1$, we have three different moments to consider with $\boldsymbol{l} = (1, 0, 0)$ and $\boldsymbol{l} = (0, 1, 0)$ and $\boldsymbol{l} = (0, 0, 1)$. We use here, in order adopt a general point of view for Eulerian spray modeling, the variable ϕ to denote the size of droplets, ϕ being the volume V the surface S or the radius R of the droplets. For the multi-fluid model, we take the surface, as mentioned in Sec. 1.2.1. The moments of the NDF define global mean spray quantities as the number density, the velocity and the temperature:

$$\begin{aligned}
M_{0\,\boldsymbol{0}\,0} &= \int_\phi \int_{\boldsymbol{u}} \int_T f^\phi(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) \, \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \\
\frac{M_{0\,\boldsymbol{1}\,0}}{M_{0\,\boldsymbol{0}\,0}} &= \frac{1}{M_{0\,\boldsymbol{0}\,0}} \int_\phi \int_{\boldsymbol{u}} \int_T \boldsymbol{u} \, f^\phi(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) \, \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \\
\frac{M_{0\,\boldsymbol{0}\,1}}{M_{0\,\boldsymbol{0}\,0}} &= \frac{1}{M_{0\,\boldsymbol{0}\,0}} \int_\phi \int_{\boldsymbol{u}} \int_T T \, f^\phi(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) \, \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T.
\end{aligned} \tag{2.13}$$

The evolution of these spray global quantities can therefore be derived from the Williams kinetic equation Eq. (1.30), in the following way:

$$\begin{aligned}
&\int_\phi \int_{\boldsymbol{u}} \int_T \quad \text{Eq. (1.30)} \quad \mathrm{d}\phi\, \mathrm{d}\boldsymbol{u}\, \mathrm{d}T, \\
&\int_\phi \int_{\boldsymbol{u}} \int_T \quad \boldsymbol{u}\,\text{Eq. (1.30)} \quad \mathrm{d}\phi\, \mathrm{d}\boldsymbol{u}\, \mathrm{d}T, \\
&\int_\phi \int_{\boldsymbol{u}} \int_T \quad T\,\text{Eq. (1.30)} \quad \mathrm{d}\phi\, \mathrm{d}\boldsymbol{u}\, \mathrm{d}T.
\end{aligned} \tag{2.14}$$

This system provides the evolution of the NDF moments $(M_{0\,\mathbf{0}\,0}, M_{0\,\mathbf{1}\,0}, M_{0\,\mathbf{0}\,1})$. The evolution of any given moment of the NDF can be obtained in the same way. Eulerian moment methods for spray consist in solving the evolution of a given set of moments of the NDF on an Eulerian grid, to characterize the global behavior of the spray. Nevertheless, one has to notice that the moment equation derivation leads to a loss of information and that, without any peculiar assumption, the system of equations for moments is not closed, even if the kinetic model was. Indeed, some quantities can not be expressed in function of the set of moments resolved; for example with regard to velocity moments, the equation for the $|\boldsymbol{l}|^{\text{th}}$ order moment introduces the moments of order $|\boldsymbol{l}|+1$. Therefore taking one first order moment $M_{0\,(100)\,0}$ introduces the second order moments $M_{0\,(110)\,0}, M_{0\,(101)\,0}, M_{0\,(200)\,0}$. Assumptions have then to be done on the form of the NDF to close the moment evolution system. For most of the Eulerian methods for spray, the derivation of the moment system, along with its associated closures can be divided into two steps. First a form for the NDF in velocity and temperature is presumed, leading to the derivation of an intermediate closed system of conservation laws, the semi-kinetic system. The second step is devoted to the size-phase space treatment, and there exist several methods introducing different types of conservation law systems and closures. However, we can also find methods using a one step moment equations derivation with quadrature-based approximations of the NDF, obtaining the unclosed moments by quadrature formula. The ideas of the derivation of such methods are raised in the following and summarized in Fig. 2.1, Fig. 2.2 and Fig. 2.3.

Presumed PDF and semi-kinetic model

The basis of several Eulerian spray resolution methods is to presume the form of the velocity and temperature distributions at the kinetic level. The NDF can thus be written:

$$f^\phi(t,\boldsymbol{x},\phi,\boldsymbol{u},T) = n^\phi(t,x,\phi)\ \psi_u(t,\boldsymbol{x},\phi,\boldsymbol{u})\ \psi_T(t,\boldsymbol{x},\phi,T), \tag{2.15}$$

where ψ_u and ψ_T are the presumed PDF in velocity and temperature, that are assumed independent of temperature and velocity, respectively. The variable n^ϕ is the number density of the spray defined by:

$$n^\phi = \int_{\boldsymbol{u}} \int_T f^\phi(t,\boldsymbol{x},\phi,\boldsymbol{u},T)\, \mathrm{d}\boldsymbol{u}\mathrm{d}T. \tag{2.16}$$

These PDF set the velocity and temperature distributions around mean values $\boldsymbol{u}_d$ and T_d:

$$f^\phi(t,\boldsymbol{x},\phi,\boldsymbol{u},T) = n^\phi(t,x,\phi)\ \psi_u(\boldsymbol{u}-\boldsymbol{u}_d(t,\boldsymbol{x},\phi))\ \psi_T(T-T_d(t,\boldsymbol{x},\phi)), \tag{2.17}$$

where $\boldsymbol{u}_d$ and T_d are the mean velocity and temperature. One has to notice that at this level, the model is still size-conditioned, i.e., the mean values depend on the size of the droplets. From this approximation, we can derive a system of moment equations for the spray taking the moments in velocity and temperature of order zero: $M_{\mathbf{0}\,0} = n^\phi$, and one: $M_{\mathbf{1}\,0} = n^\phi\,\boldsymbol{u}_d$ and $M_{\mathbf{0}\,1} = n^\phi\,T_d$, in the following way:

$$\begin{aligned}
&\int_{\boldsymbol{u}}\int_T \quad \text{Eq. (1.30)} \quad \mathrm{d}\boldsymbol{u}\,\mathrm{d}T,\\
&\int_{\boldsymbol{u}}\int_T \quad \boldsymbol{u}\,\text{Eq. (1.30)} \quad \mathrm{d}\boldsymbol{u}\,\mathrm{d}T,\\
&\int_{\boldsymbol{u}}\int_T \quad h_l(T)\,\text{Eq. (1.30)} \quad \mathrm{d}\boldsymbol{u}\,\mathrm{d}T.
\end{aligned} \tag{2.18}$$

It gives the following semi-kinetic system:

$$\begin{aligned}
\partial_t\, n^\phi + \partial_{\boldsymbol{x}}\cdot(n^\phi \boldsymbol{u}_d) &= \partial_\phi(n^\phi \tilde{R}_\phi) + \Gamma_1 + Q_1,\\
\partial_t\,(n^\phi \boldsymbol{u}_d) + \partial_{\boldsymbol{x}}\cdot(n^\phi \boldsymbol{u}_d\otimes\boldsymbol{u}_d + \boldsymbol{P}) &= \partial_\phi(n^\phi \tilde{R}_\phi \boldsymbol{u}_d) + n^\phi\tilde{\boldsymbol{F}} + \Gamma_2 + Q_2,\\
\partial_t\, n^\phi \tilde{h}_l + \partial_{\boldsymbol{x}}\cdot(n^\phi \tilde{h}_l \boldsymbol{u}_d) &= \partial_\phi(n^\phi \tilde{h}_l \tilde{R}_\phi) + n^\phi C_{p,l}\tilde{E} + \Gamma_3 + Q_3,
\end{aligned} \tag{2.19}$$

where h_l is the droplet enthalpy. The equation was derived for the enthalpy, instead of temperature since it is the conservative variable. The notation $\tilde{(.)}$, define the mean quantities:

$$\tilde{g}(t,\boldsymbol{x},\phi) = \int_{\boldsymbol{u}}\int_T g(t,\boldsymbol{x},\phi,\boldsymbol{u},T)\,\psi_u\left(\boldsymbol{u}-\boldsymbol{u}_d(t,\boldsymbol{x},\phi)\right)\,\psi_T\left(T-T_d(t,\boldsymbol{x},\phi)\right)\,\mathrm{d}\boldsymbol{u}\,\mathrm{d}T. \tag{2.20}$$

The droplet interaction terms, for collisions or coalescence, and the secondary breakup terms, are given by:

$$\begin{aligned}
\mathrm{I}_1(n^\phi)(t,\boldsymbol{x},\phi) &= \int_{\boldsymbol{u}}\int_T \mathrm{I}(f^\phi)(t,\boldsymbol{x},\phi,\boldsymbol{u},T)\,\mathrm{d}\boldsymbol{u}\mathrm{d}T,\\
\mathrm{I}_{\mathbf{2}}(n^\phi)(t,\boldsymbol{x},\phi) &= \int_{\boldsymbol{u}}\int_T \boldsymbol{u}\,\mathrm{I}(f^\phi)(t,\boldsymbol{x},\phi,\boldsymbol{u},T)\,\mathrm{d}\boldsymbol{u}\mathrm{d}T,\\
\mathrm{I}_3(n^\phi)(t,\boldsymbol{x},\phi) &= \int_{\boldsymbol{u}}\int_T h_l(T)\,\mathrm{I}(f^\phi)(t,\boldsymbol{x},\phi,\boldsymbol{u},T)\,\mathrm{d}\boldsymbol{u}\mathrm{d}T,
\end{aligned} \tag{2.21}$$

I standing for Γ or Q. $\boldsymbol{P}$ is a tensor equivalent to a pressure term, describing the velocity dispersion of the droplets:

$$\boldsymbol{P} = \int_{\boldsymbol{u}}\int_T n^\phi\,\psi_u\left(\boldsymbol{u}-\boldsymbol{u}_d\right)\,\psi_T\left(T-T_d\right)\,\left(\boldsymbol{u}-\boldsymbol{u}_d\right)\otimes\left(\boldsymbol{u}-\boldsymbol{u}_d\right)\mathrm{d}\boldsymbol{u}\,\mathrm{d}T. \tag{2.22}$$

One has to notice that the evolution of number density and of mass density are equivalent as we assumed the liquid density ρ_l independent of temperature:

$$m^\phi = \rho_l\,Vol(\phi)\int_{\boldsymbol{u}}\int_T f^\phi(t,\boldsymbol{x},\phi,\boldsymbol{u},T)\,\mathrm{d}\boldsymbol{u}\,\mathrm{d}T, \tag{2.23}$$

where $Vol(\phi)$ is the volume of the droplet. Its expression depends on the choice of the size variable ϕ:

$$Vol(\phi) = V = \frac{S^{3/2}}{6\sqrt{\pi}} = \frac{4\pi\, R^3}{3}. \tag{2.24}$$

In the DNS framework, the function ψ_u and ψ_T are taken as Dirac delta-functions:

$$f^{\phi}(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) = n^{\phi}(t, x, \phi)\ \delta(\boldsymbol{u} - \boldsymbol{u}_d(t, \boldsymbol{x}, \phi))\ \delta(T - T_d(t, \boldsymbol{x}, \phi)). \tag{2.25}$$

In this context, there is no velocity dispersion around the mean value and $\boldsymbol{P} = 0$. Furthermore, the quantities $\tilde{(.)}$ are the point-wise values of the functions at $\boldsymbol{u} = \boldsymbol{u}_d$ and $T = T_d$:

$$\begin{array}{rclcrcl} \tilde{R_\phi} & = & R_\phi(\boldsymbol{u}_d, T_d) & = & R_{\phi d}, & \qquad \tilde{\boldsymbol{F}} & = & \boldsymbol{F}(\boldsymbol{u}_d) & = & \boldsymbol{F}_d, \\ \tilde{h_l} & = & h_l(\boldsymbol{u}_d, T_d) & = & h_{ld}, & \qquad \tilde{E} & = & E(\boldsymbol{u}_d, T_d) & = & E_d. \end{array} \tag{2.26}$$

Size-phase space treatment still remains to be defined. Two types of approach can be distinguished:

- whole size-phase space integration, leading to two-fluid type models,
- size-phase space discretization, leading to multi-fluid type models.

The derivation of these kind of methods is summarized in Fig. 2.2.

Quadrature based NDF approximation

Another way to close the system of equations for moment evolution is to use quadrature formula to obtain the unknown high order moments as a function of low order moments. This method called Quadrature methods of moments, (McGraw 1997; Marchisio, Vigil, and Fox 2003), is unherited from moment methods for fluid-particle systems derived from population balance equation, reviewed in (Ramkrishna and Fredrickson 2000). This method was derived for cases with univariate distribution function, $g(t, x, u)$, with one internal coordinate, i.e., for one-dimensional case, and where the evolution of the k^{th} order moment is given by:

$$\partial_t M_k + \partial_x(u_d M_k) = \int_u Q(u) g(t, x, u)\, \mathrm{d}u. \tag{2.27}$$

In this case the closure is given by the quadrature formula:

$$\int_u Q(u) g(t, x, u)\, \mathrm{d}u = \sum_{n=1}^{N} w_n(t, x)\, Q(u_n(t, x)), \tag{2.28}$$

with the weights w_n and the abscissas u_n. It corresponds to assume for the distribution function the following shape:

$$g(t, x, u) = \sum_{n=1}^{N} w_n(t, x) \delta(u - u_d(t, x)). \tag{2.29}$$

The system is closed as these weights and abscissas are linked to the low order moments through the product difference (PD) algorithm, (McGraw 1997):

$$\int_u u^k \, g \, \mathrm{d}u = \sum_{n=1}^{N} w_n u_n^k. \tag{2.30}$$

The great advantage of this method is its ability to describe particle trajectory crossings (PTC). Indeed when particles are crossing, there are two different velocities at the same point, and presumed velocity PDF presented in the previous paragraph, as they describe an equilibrium distribution around a mean velocity, cannot describe this out of equilibrium case. A third order QMOM method, using $2^{N_{\mathrm{dim}}}$ quadrature nodes, N_{dim} being the dimension of the domain, can deal with configurations where PTC occur, (Fox 2008). Nevertheless, this method reduces to univariate distribution function and can not therefore be applied directly to the spray simulation where distribution functions have at least two (ϕ, u) or three (ϕ, u, T) internal coordinates.

However, an extension of QMOM method has been proposed in (Marchisio and Fox 2005), to treat distribution functions with several internal coordinates. In this case the approximation of the shape of the distribution is extended from QMOM for multi-variate distribution, (Marchisio and Fox 2005):

$$f^{\phi}(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) = \sum_{n=1}^{N} w_n \, \delta(\phi - \phi_n) \, \delta(\boldsymbol{u} - \boldsymbol{u}_n) \, \delta(T - T_n), \tag{2.31}$$

where N is the number of Dirac delta-functions and w_n is the weight of the node n. One has to notice that the weights w_n, and abscissas $(\phi_n, \boldsymbol{u}_n, T_n)$, are Eulerian fields. The moments of the distribution are thus given by:

$$M_{k\boldsymbol{l}m} = \int_{\phi}\int_{\boldsymbol{u}}\int_{T} \phi^k \, \boldsymbol{u}^{\boldsymbol{l}} \, T^m \, f^{\phi}(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) \mathrm{d}\phi \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T = \sum_{n=1}^{N} w_n \, \phi_n^k \, \boldsymbol{u}_n^{\boldsymbol{l}} \, T_n^m, \tag{2.32}$$

where $M_{k\boldsymbol{l}m}$ is the moment of order k in size, $|\boldsymbol{l}|$ in velocity and m in temperature. The main difference with the QMOM method is that instead of solving the evolution of the moment, it directly solves the evolution of the weights and abscissas of the quadrature. This method, called Direct Quadrature Method of Moments (DQMOM) in (Marchisio and Fox 2005), is an alternative to the presumed PDF method for Eulerian spray description. The derivation of QMOM and DQMOM method is summarized in Fig. 2.3. The details concerning these methods can be found in (Fox 2007).

2.2.2 Two-fluid model

Two-fluid derivation

The two-fluid model, for dispersed liquid case framework, is based on a presumed PDF in velocity and temperature. This model is studied in (Drew and Passman 1999; Chanteperdrix, Villedieu, and Vila 2002; Murrone and Guillard 2005). It can be derived from the semi-kinetic model, realizing an integration over the whole size phase space, see Fig. 2.2:

$$\int_{\phi} \text{Eq. (2.19)} \, \mathrm{d}\phi, \tag{2.33}$$

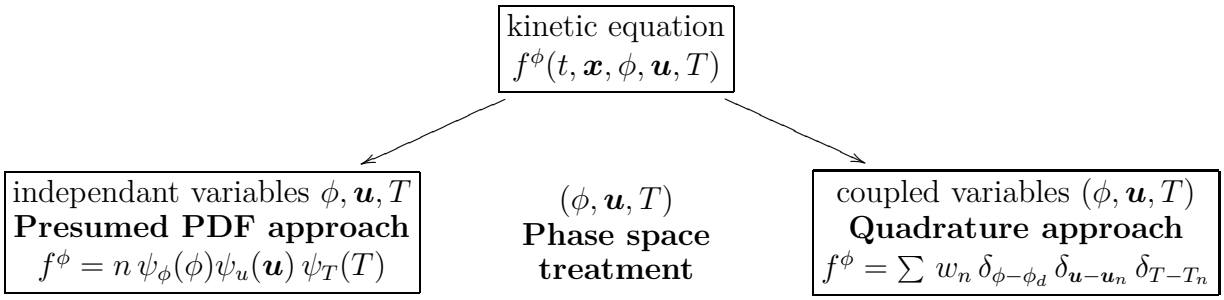


Figure 2.1 – *First step for moment method derivation from the kinetic description, Eq. (1.30): velocity-temperature phase space resolution of the spray NDF*

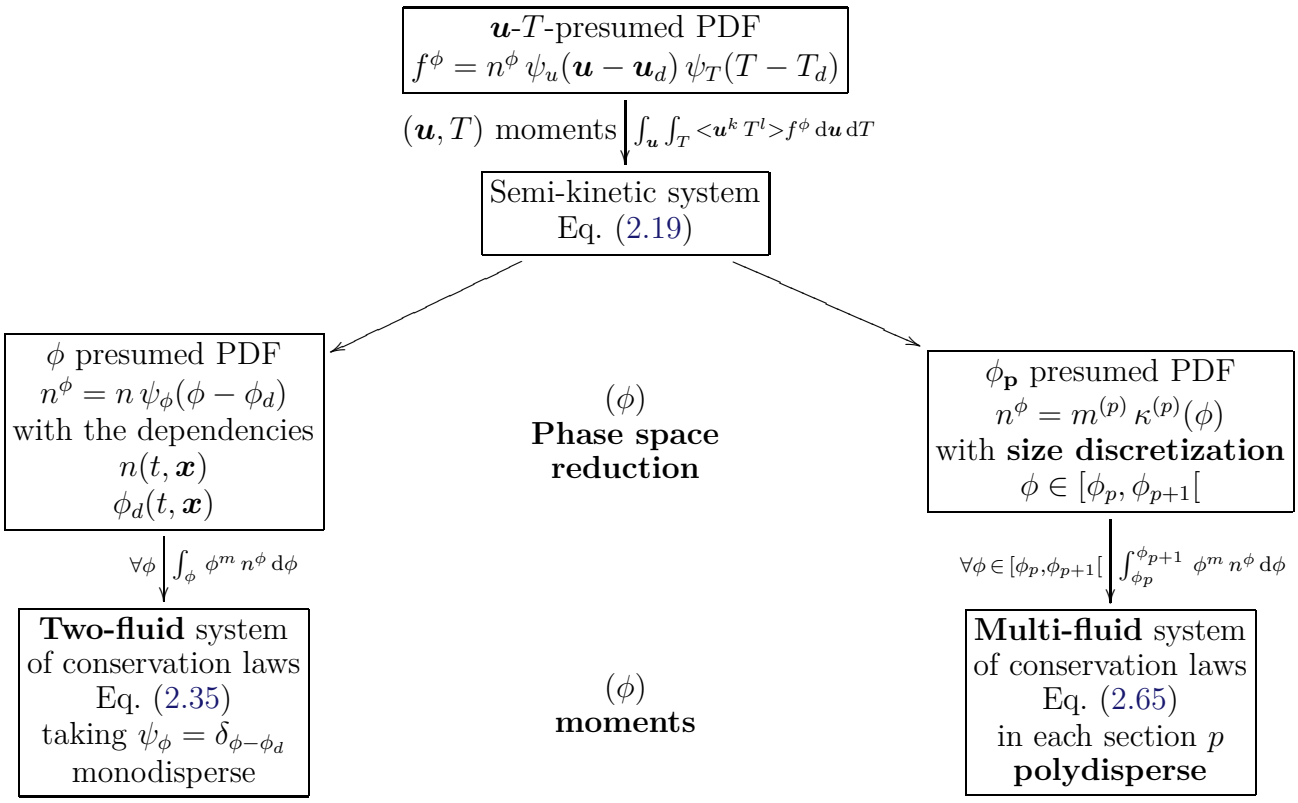


Figure 2.2 – *Velocity and temperature presumed PDF method derivation. Size phase space closure and moment derivation for two-fluid methods and "classical" multi-fluid method.*

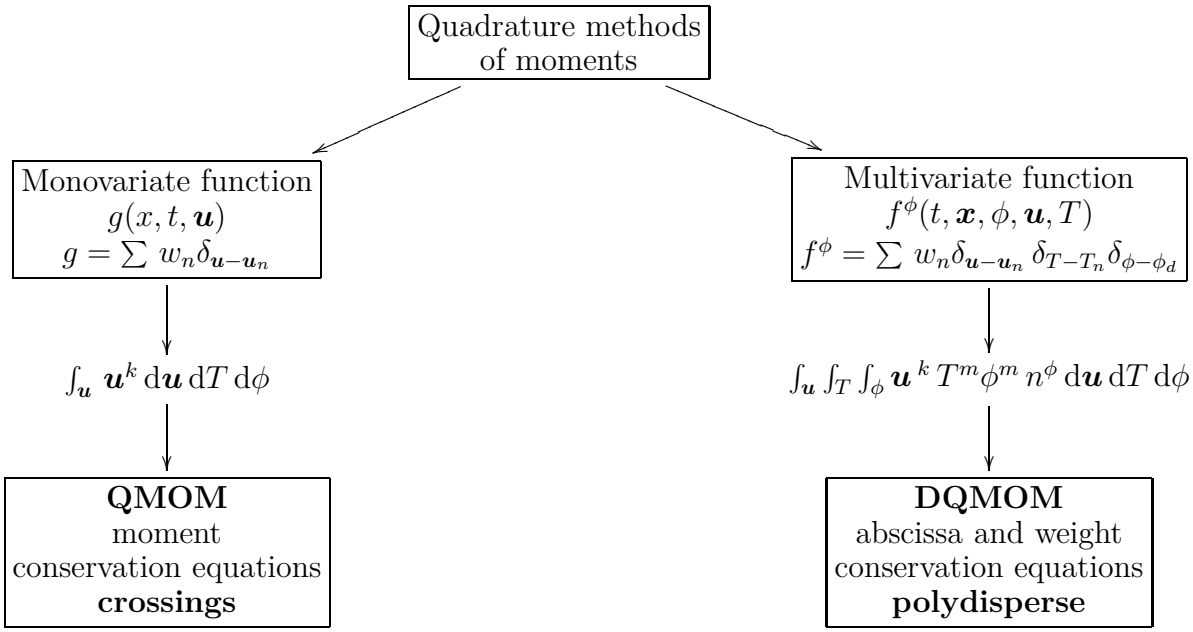


Figure 2.3 – *Velocity and temperature quadrature based method derivation. Size phase space closure and moment derivation for QMOM/DQMOM and multi-fluid multi-velocity methods.*

assuming, (Dufour 2005):

$$n^\phi(t,\boldsymbol{x},\phi) = n(t,\boldsymbol{x})\delta(\phi - \phi_d(t,\boldsymbol{x})). \tag{2.34}$$

It gives the following evolution for the number and the mass density, the momentum and the enthalpy:

$$\begin{aligned}
\partial_t n + \partial_{\boldsymbol{x}}\cdot(n\boldsymbol{u}_d) &= 0,\\
\partial_t(\alpha\rho_l) + \partial_{\boldsymbol{x}}\cdot(\alpha\rho_l\boldsymbol{u}_d) &= -\dot{m}_p,\\
\partial_t(\alpha\rho_l\boldsymbol{u}_d) + \partial_{\boldsymbol{x}}\cdot(\alpha\rho_l\boldsymbol{u}_d\otimes\boldsymbol{u}_d) &= -\beta\alpha\rho_l(\boldsymbol{u}_d - \boldsymbol{U}_{\boldsymbol{g}}) - \dot{m}_p\boldsymbol{u}_d,\\
\partial_t(\alpha\rho_l h_{ld}) + \partial_{\boldsymbol{x}}\cdot(\alpha\rho_l h_{ld}\boldsymbol{u}_d) &= Q_l - \dot{m}_p h_{ld},
\end{aligned} \tag{2.35}$$

where α is the liquid volume fraction, $\dot{m}_p$ is the mass flux due to evaporation, β is the drag coefficient and Q_l represents the heat transfer. In this framework, a mean size can be defined for the droplets by the following formula:

$$R = \left(\frac{3\alpha}{4\pi n}\right)^{1/3}. \tag{2.36}$$

Nevertheless, the information on the size distribution and on the size-velocity correlation is lost in this derivation. Closures are thus needed for the terms $\dot{m}_p$, α and Q_l. In this framework, the droplet interaction terms, needed to account for secondary breakup, collisions and coalescence, cannot be included directly but can only be treated by rough approximations.

Polydisperse extension

In order to obtain a more precise description of the size distribution and to take into account droplet interactions, two methods have been developed to extend the two-fluid model.

Sampling method A first method is to realize a sampling of the distribution and to define N samples by:

$$f^{\phi}(t,\boldsymbol{x},\phi,\boldsymbol{u},T)=\sum_{i=1}^{N} n_i(t,\boldsymbol{x})\delta(\phi-\phi_{di}(t,\boldsymbol{x}))\delta(\boldsymbol{u}-\boldsymbol{u}_{di}(t,\boldsymbol{x}))\delta(T-T_{di}(t,\boldsymbol{x})), \quad (2.37)$$

each sample having its mean size, ϕ_{di}, velocity, $\boldsymbol{u}_{di}$, and temperature T_{di}. The system of equations for each sample is derived in the way used for the classical two-fluid method, presuming the shapes of the velocity, temperature and size distributions. It thus results in N systems of equations similar to Eq. (2.35). This sampling is illustrated in Fig. 2.4. Nevertheless, one has to notice that the sampled sizes are not interacting and thus the

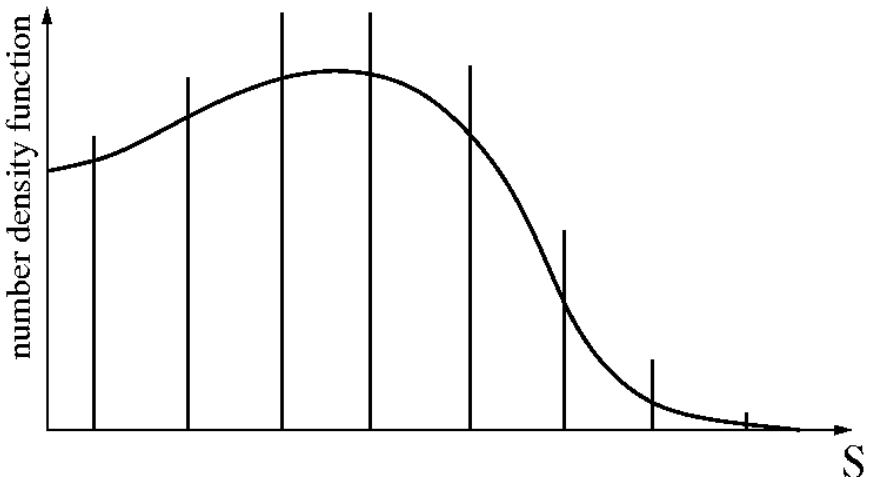


Figure 2.4 – *Size distribution sampling realized to derive sampling methods.*

description of the phenomena inducing coupling in the size phase space, as coalescence or secondary breakup, are not taken into account.

Presumed size distribution Another way to obtain a description of the size distribution is to presume a PDF for the size variable, as done in (Babinsky and Sojka 2002; Mossa 2005). The idea is to presume a form for the size distribution and to solve the evolution of a set of moments of this distribution. Nevertheless, this approach suffers from severe problem coming from both the evaporation process and the coupling with the velocity conditioned by size. Indeed it is necessary to presume a form of the spray NDF, for instance a log-normal form, and to resolve the evolution of this log-normal parameters. Nevertheless, the form of the NDF is not conserved through evaporation: an initially log-normal distribution evaporating has no reason to stay a log-normal distribution. Therefore severe singularities occur in the equations governing the evolution of the log-normal parameters.

2.2.3 Multi-fluid model

The multi-fluid method, introduced in (Greenberg, Silverman, and Tambour 1993) and characterized in (Laurent and Massot 2001), proposes to conserve information on the size distribution as well as on the size-velocity correlations, discretizing the size phase space. The semi-kinetic system Eq. (2.19) is averaged on fixed size intervals $[\phi_p, \phi_{p+1}[, p \in [1, N_S]$ called sections, where N_S is the number of sections. We thus solve the evolution of moments of the NDF in each section, the sections being coupled by exchange of mass, momentum and heat, see Fig. 2.5, Fig. 2.2. For each section p, a system of conservation law for the

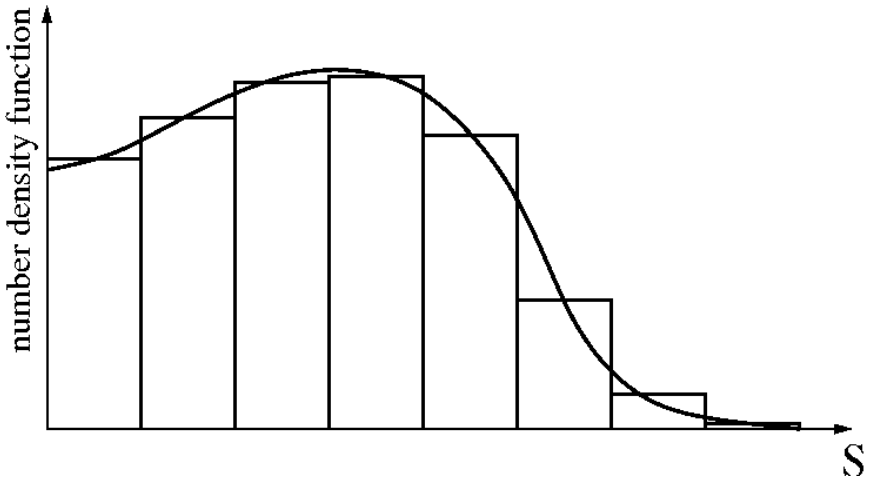


Figure 2.5 – *Size distribution function discretization.*

mass, the momentum and the enthalpy, given by:

$$\begin{aligned} m^{(p)}(t,\boldsymbol{x}) &= \int_{\phi_p}^{\phi_{p+1}} \rho_l \, Vol(\phi) \, n^{\phi}(t,\boldsymbol{x},\phi) \, \mathrm{d}\phi, \\ m^{(p)}(t,\boldsymbol{x}) \, \boldsymbol{u}_d^{(p)}(t,\boldsymbol{x}) &= \int_{\phi_p}^{\phi_{p+1}} \rho_l \, Vol(\phi) \, n^{\phi}(t,\boldsymbol{x},\phi) \, \boldsymbol{u}_d(t,x,\phi) \, \mathrm{d}\phi, \\ m^{(p)}(t,\boldsymbol{x}) \, h_{l_d}^{(p)}(t,\boldsymbol{x}) &= \int_{\phi_p}^{\phi_{p+1}} \rho_l \, Vol(\phi) \, n^{\phi}(t,\boldsymbol{x},\phi) \, h_{ld}(t,x,\phi) \, \mathrm{d}\phi, \end{aligned} \tag{2.38}$$

is derived from the semi-kinetic system:

$$\int_{\phi_p}^{\phi_{p+1}} \rho_l \, Vol(\phi) \text{ Eq. (2.19) } \mathrm{d}\phi. \tag{2.39}$$

The detailed equations for the multi-fluid model are given in Sec. 2.3, and we just give here the key elements concerning its derivation. As far as closures are concerned, the size phase space discretization allows to take into account evaporation or droplet interactions phenomena coupling droplets of different sizes. The number distribution in one section, $n^{\phi}(t,\boldsymbol{x},\phi), \phi \in [\phi_p, \phi_{p+1}[$, is unclosed as we only solve one moment in size of this distribution. We thus presume the profile of n^{ϕ} in a given section, as a function only of the droplet geometry:

$$n^{\phi}(t,x,\phi) = m^{(p)}(t,x) \, \kappa^{(p)}(\phi). \tag{2.40}$$

The size distribution function is generally approximated by a piecewise constant function, $\kappa^{(p)}$. The form of $\kappa^{(p)}$ depends of the choice for ϕ and will be different for V, S and R. This is known as the fundamental sectional assumption, (Greenberg, Silverman, and Tambour 1993; Laurent and Massot 2001). Unlike the presumed size distribution method, the form of the size distribution is assumed only within a section in the multi-fluid model. This model thus describes the polydispersion as well as the size velocity correlation, a key issue in evaporating spray simulations. However, the main drawback of this method is the cost associated to the resolution of N_S systems of conservation laws, needed to obtain the evolution of the sections. Indeed, it has been shown that the multi-fluid model is a first order method regarding evaporation, (Laurent 2006), and thus a high number of sections, typically ten to twenty, may be needed to obtain a precise description of the evaporation process. This drawback can be avoided by using methods of higher order for the description of evaporation in multi-fluid framework. A second important limitation, comes from the description of droplet crossing trajectories, that can not be done with standard multi-fluid method, as with all presumed velocity PDF based Eulerian method. Nevertheless a new method coupling multi-fluid method with QMOM, the multi-fluid multi-velocity model, has been proposed to answer to this problem. These new outbreaking methods are developed in the following and summarized in Fig. 2.6.

Higher order method for evaporation

A first way to increase multi-fluid precision regarding size description is to assume for n^ϕ a more complex form than the piecewise constant form defined by Eq. (2.40). While the form of n^ϕ was only depending on size

$$n^\phi(\phi) = \psi^{(p)}(\phi),\ \phi \in [\phi_p, \phi_p[, \tag{2.41}$$

the use of a more complex form would introduce new parameters:

$$n^\phi(\phi) = \psi^{(p)}(a, b, \phi),\ \phi \in [\phi_p, \phi_p[. \tag{2.42}$$

A new conservation equation is needed to close the system, and it is obtained taking a second moment in size of the distribution function. Indeed the classical multi-fluid was solving only one moment of the size distribution, the mass $m^{(p)}$. This technique has been applied in (Dufour and Villedieu 2005), where a piecewise exponential form is chosen for the size distribution, using surface variable:

$$n(t, \boldsymbol{x}, S) = a^{(p)}(t, \boldsymbol{x}) \exp(-b^{(p)}(t, \boldsymbol{x})S),\ S \in [S_p, S_{p+1}[. \tag{2.43}$$

As previously mentioned, we choose the surface as the size variable for the multi-fluid formulation, (Laurent 2006). In order to close this formulation, two moments, of order 0 and 3/2 in surface, of the size distribution function are solved:

$$\int_{S_p}^{S_{p+1}} \begin{pmatrix} 1 \\ S^{3/2} \end{pmatrix} \text{Eq. (2.19)}\ \mathrm{d}S. \tag{2.44}$$

The moment of order 3/2 in surface is already solved in the standard multi-fluid, as it corresponds to the mass, as $Vol(S) = S^{3/2}/(6\sqrt{\pi})$ and ρ_l is constant. A new equation is added for the conservation of the moment of order zero, corresponding to the conservation

of the number density. In this case, the conservation of mass and of number density are not equivalent. The details of the multi-fluid derivation in this framework can be found in (Dufour 2005) and (Dufour and Villedieu 2005)

Another form has been proposed for the number density inside one section in (Laurent 2006). The distribution is assumed piecewise polynomial. The idea here is to solve a higher number of moments m of the size distribution function to increase the order of precision of the method. The method is assessed for $m = 2$ with moments of order 0 and 1 in surface:

$$\int_{S_p}^{S_{p+1}} \begin{pmatrix} 1 \\ S \end{pmatrix} \text{ Eq. (2.19) } \mathrm{d}S. \tag{2.45}$$

All the details concerning the derivation of the multi-fluid model in this framework along with the numerical method needed are given in (Laurent 2006).

These methods achieve second order precision in the size variable. Nevertheless, a new method for the size phase space treatment have been recently proposed, (Kah 2007; Massot, Laurent, Kah, and de Chaisemartin 2009), leading to an important improvement in precision. The method solves the evolution of four moments in size of order 0 to 3, in each section p:

$$\int_{S_p}^{S_{p+1}} \begin{pmatrix} 1 \\ S \\ S^2 \\ S^3 \end{pmatrix} \text{ Eq. (2.19) } \mathrm{d}S. \tag{2.46}$$

The closure is based on a reconstruction of the number density n from the four moments solved. It allows to evaluate the point-wise value of the distribution at the boundary of the sections to compute the flux between the sections of the multi-fluid model. This reconstruction step must satisfy two main constraints:

- preserve positivity,
- preserve the moment space.

The second constraint consists in ensuring that the vector of moments modified by the fluxes of the method stays a vector of moments of a distribution function. To satisfy these constraints, a maximum entropy formalism as well as a kinetic scheme are used, as detailed in (Massot, Laurent, Kah, and de Chaisemartin 2009). The entropy comes from information theory. A particular attention must also be devoted when dealing with the evolution in physical space of the vector of moments, again for moment space preservation purpose, (Kah 2007). This recently developed method leads to a great improvement of the precision, and allows a very precise description of the size distribution with only few size sections.

Multi-fluid multi-velocity model

An important limitation arising with Eulerian methods based on a presumed velocity distribution around a mean velocity, is the impossibility to model droplet trajectory crossings. A recent model coupling multi-fluid model and QMOM, see Fig. 2.6, has been proposed in (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009) and (Freret,

de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009). The derivation of this method differs from the standard multi-fluid model one. Indeed, there is no intermediate step associated with a semi-kinetic system based on a presumed velocity PDF. The form of the spray NDF $f(t, \boldsymbol{x}, S, \boldsymbol{u}, T)$ is directly assumed within a multi-fluid section p:

$$f(t, \boldsymbol{x}, S, \boldsymbol{u}, T) = n^{(p)}(t, \boldsymbol{x}, S)\psi_u^{(p)}(t, \boldsymbol{x}, \boldsymbol{u}), \tag{2.47}$$

where $n^{(p)}$ is the size number density function and $\psi_u^{(p)}$ is the velocity probability density function, which is assumed independent of size variation inside the p^{th} section. We still consider the mass moment of order 3/2 in size, zero or one in temperature, depending whether heat transfer are considered, and up to order three in velocity. The moments considered in the p^{th} section can be written:

$$m^{(p)}\, M^{(p)}_{\frac{3}{2}\,\boldsymbol{l}\,k} = \int_{S_p}^{S_{p+1}} \rho_l\, S^{3/2}\, n^{(p)}\, \mathrm{d}S \int_{\boldsymbol{u}} \boldsymbol{u}^{\boldsymbol{l}}\, \psi_u^{(p)}(t, \boldsymbol{x}, \boldsymbol{u})\, \mathrm{d}\boldsymbol{u} \int_T T^k \psi_T^{(p)}(t, \boldsymbol{x}, \boldsymbol{u})\, \mathrm{d}T, \tag{2.48}$$

with $|\boldsymbol{l}| \leq 3$ for a 3-D case and k equal to zero or one. The transport in physical space is then closed with a quadrature methods of moments extended from (Fox 2008). In order to be able to treat trajectory crossings, a $2^{N_{\text{dim}}}$-node quadrature is used, the moments being linked to the weights w_n and abscissa $\boldsymbol{u}_n$ by:

$$M^{(p)}_{\frac{3}{2}\,\boldsymbol{l}\,k} = \sum_{n=1}^{2^{N_{\text{dim}}}} w_n \left(\boldsymbol{u}_n^{(p)}\right)^{|\boldsymbol{l}|}. \tag{2.49}$$

The difficulty is to obtain the weights and abscissas from the moments, the resolution of this task is described in (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009).

A method treating the size polydispersion as well as droplet crossings for evaporating, (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009), and colliding spray, (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009), has thus been developed and assessed in 2-D configurations.

2.2.4 Direct Quadrature Method Of Moments for spray

As we mentioned in Sec. 2.2.1, an alternative to the presumed velocity PDF Eulerian method for the simulation of spray is the Direct Quadrature Method of Moments (DQMOM), introduced in (Marchisio and Fox 2005). In this framework, the NDF of the spray is assumed to have the form:

$$f^{\phi}(t, \boldsymbol{x}, \phi, \boldsymbol{u}, T) = \sum_{n=1}^{N} w_n\, \delta(\phi - \phi_n)\, \delta(\boldsymbol{u} - \boldsymbol{u}_n)\, \delta(T - T_n), \tag{2.50}$$

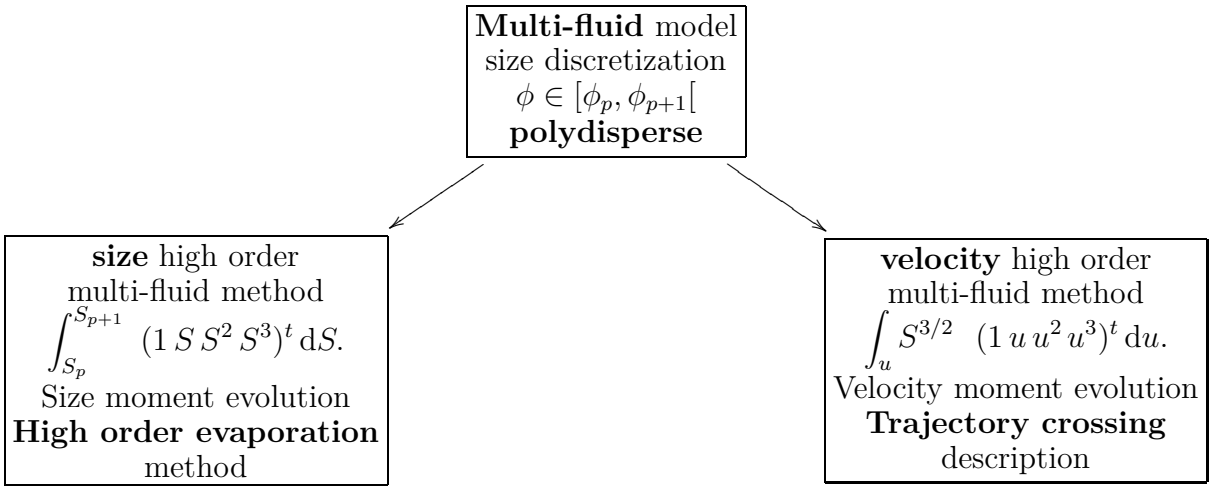


Figure 2.6 – *Size/Velocity high order multi-fluid method. These methods describe accurately droplet evaporation and trajectory crossings, respectively.*

and a system of conservation laws is derived directly for the weights and abscissas $(w_n, \phi_n, \boldsymbol{u}_n, T_n)$:

$$\begin{aligned} \partial_t w_n + \partial_{\boldsymbol{x}} \cdot (w_n \boldsymbol{u}_n) &= a_n, \\ \partial_t (w_n \phi_n) + \partial_{\boldsymbol{x}} \cdot (w_n \phi_n \boldsymbol{u}_n) &= b_n, \\ \partial_t (w_n \phi_n \boldsymbol{u}_n) + \partial_{\boldsymbol{x}} \cdot (w_n \phi_n \boldsymbol{u}_n \otimes \boldsymbol{u}_n) &= c_n, \\ \partial_t (w_n T_n) + \partial_{\boldsymbol{x}} \cdot (w_n T_n \boldsymbol{u}_n) &= d_n, \end{aligned} \tag{2.51}$$

where the source terms (a_n, b_n, c_n, d_n) are obtained from the source terms of the Williams equation Eq. (1.30); the evolution of the weights and abscissas allows to compute the moments and the moments allow to compute the source terms, (Marchisio and Fox 2005), solving a linear system of $6N$ unknowns.

The DQMOM method, along with the multi-fluid method, have been compared to a Lagrangian description in the framework of a 1-D vaporizing and coalescing laminar spray (Fox, Laurent, and Massot 2008). It has been shown that the DQMOM method, first introduced in the context of aerosol dynamics and aggregation-breakage processes, can be implemented for a vaporizing spray described by a Williams equation, Eq. (1.30), as also tackled in (Fox 2007). The DQMOM reveals to be very efficient to describe spray coalescence, more efficient that the multi-fluid due to its limited numerical diffusion in the size phase space. As far as evaporation is concerned, the method is shown to be efficient but the importance to treat precisely the problem of the evaporative flux due to droplets disappearance is highlighted. An efficient solution to this problem, common to several moment methods for fluid-particle flows, is tackled in (Massot, Laurent, Kah, and de Chaisemartin 2009).

Finally, up to now, the DQMOM fails to treat droplet trajectory crossing, (Fox 2008).

2.2.5 Extension to the simulation of turbulent flows

A DNS framework for the Eulerian moment methods derivation has been considered so far. The extension to modeled or partially modeled flow simulations is assessed here. We do not tackle the gas phase description, but we discuss the effect of a gas phase partially modeled, on the Eulerian models derivation. In a Large Eddy Simulation (LES) context, small scales of the gas phase are modeled. In this case, the turbulence description alters the kinetic level of description, and the kinetic equation presented in Sec. 1.2 is no longer closed, (Reeks 1991).

Two-phase turbulent flows modeling and LES description

In order to assess more realistic configurations, the gas phase needs to be, at least partially, modeled. In the context of LES, the information on the gas velocity is only partial, the small scales being modeled, and the kinetic formulation for spray needs thus to be adapted. The strategy is to start at the kinetic level with a realization of the Williams equation Eq. (1.30) and to derive a kinetic equation in the average where the effects of the small scales of the turbulent gas phase are modeled. The idea is to obtain a closed equation at the kinetic level and then, from there, at the fluid level. This approach, as well as its main interests are described originally in (Reeks 1991).

Kinetic model "in the mean" The derivation of a transport equation for the NDF of droplets in turbulent flows has been the subject of many researchers in the past two decades. We do not detail here the various works and we refer to the reviews done in (Mashayek and Pandya 2003; Minier and Peirano 2001; Kaufmann 2004). We only recall here the form of the NDF's equation obtained in (Reeks 1991; Reeks 1992; Reeks 1993) and (Hyland, McKee, and Reeks 1999). The carrier velocity is split into a mean and a fluctuating part:

$$U_g(t,\boldsymbol{x}) = \langle U_g\rangle(t,\boldsymbol{x}) + U_g'(t,\boldsymbol{x}), \tag{2.52}$$

where $\langle . \rangle$ denotes the ensemble averaging operator with respect to the randomness introduced through the fluctuating quantity. A kinetic equation "in the mean" can be obtained by ensemble average of the Williams equation, to describe the evolution of $\langle f\rangle$, given, if no droplet interactions are considered by:

$$\partial_t\langle f\rangle + \partial_{\boldsymbol{x}}\cdot(\boldsymbol{u}\langle f\rangle) + \partial_{\boldsymbol{u}}\cdot\left(\langle \boldsymbol{F}\rangle\langle f\rangle + \left\langle \frac{U_g'}{\mathrm{St}} f\right\rangle\right) - \partial_S\left(R_s\,\langle f\rangle\right) + \partial_T\left(E\langle f\rangle\right) = 0. \tag{2.53}$$

In this formulation, it is assumed that no convective corrections are conducted for evaporation and heat transfer, so that the droplet size and temperature change rates, R_s and E, are not modified by the stochastic process. One has to notice that the term $\langle U_g' f\rangle$ is not closed and we refer to (Reeks 1992) for a closure proposition. This closure, along with an equilibrium assumption on the velocity distribution conditioned by size, allow us to derive, as in Sec. 2.2.1, a semi-kinetic system of conservation laws, taking in this case the moment of order 0, 1 and 2 in velocity, see for example (Massot 2007). Two-fluid and multi-fluid model derivation can then be conducted from this point, with their associated size phase space treatment, see Sec. 2.2.2 and Sec. 2.2.3.

LES and two-fluid model As far as two-fluid models are concerned, many efforts have been devoted to a derivation for LES framework, and this method is, at present, the most advanced for LES of turbulent flow. The basis of turbulent modeling are given in (Fevrier 2000), and the first formulation of two-phase flow LES is to be found in (Kaufmann 2004). Many works were first devoted to LES of solid particles, (Fevrier 2000; Fevrier, Simonin, and Squires 2005; Moreau 2006; Riber 2007), and completed by studies on monodisperse spray in reacting flows, (Riber, Garcia, Moureau, Pitsch, Simonin, and Poinsot 2006; Boileau 2007; Lamarque 2007). An extension of LES for polydisperse spray is proposed in (Mossa 2005), with a size presumed PDF, Sec. 2.2.2. The LES filtering for the equations obtained from the ensemble average formulation is described in (Kaufmann 2004). The filtered equations for the two-fluid described disperse phase are developed in (Kaufmann 2004), and modeling of unresolved terms for the disperse phase LES is detailed in (Kaufmann 2004; Moreau 2006; Riber 2007).

Toward LES with multi-fluid approach Two issues have to be tackled to extend the multi-fluid method used in this study to LES of turbulent flow.
The first point for LES with the multi-fluid method is to provide size-conditioned closures. First size-conditioned closures for turbulent approach are studied in (Reveillon, Péra, and Massot 2002; Massot, Knikker, Péra, and Reveillon 2004; Reveillon, Péra, Massot, and Knikker 2004). A multi-fluid model is derived to achieve ensemble average simulation (EAS) of sprays in turbulent flow fields. The efficiency of the multi-fluid is assessed through a comparison to a Euler-Lagrange DNS of a non homogeneous turbulent two-phase flow. A spatially decaying turbulence is used, offering a spatial evolution of stationary statistics. The study uses one-way coupling and focuses on the sole effects of the turbulence fluctuations on the spray dispersion. The ensemble averaged Eulerian fields obtained from the Eulerian multi-fluid simulation are compared to the ones analyzed from the Euler-Lagrange DNS. This study shows the ability to capture the physics, see the references above for more details, and provides a first step toward LES simulation with multi-fluid.
Second, mathematical studies of the resulting equations could provide the elements to extend the numerical scheme provided in this work, to this framework. The main difference comes from the pressure arising in the momentum equation, vacant in the DNS case, due to the monokinetic assumption.

2.3 Monokinetic multi-fluid model detailed derivation

In the present study, DNS with multi-fluid spray description is provided, in Part. II and Part. IV. The "classical" multi-fluid model is used, that corresponds to a low order moment method in size and velocity, coupled with a size discretization. This method is chosen given its relative simplicity and robustness, an essential issue to assess the more realistic configurations tackled in Part. IV.

Recent improvements have been obtained for the multi-fluid method, Sec. 2.2.3. Indeed, a high-order multi-fluid moment method in size, (Massot, Laurent, Kah, and de Chaisemartin 2009), has been provided for a refined evaporation description. Besides, a high-order multi-fluid moment method in velocity (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009; Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon,

and Massot 2009), allows to describe out-of-equilibrium velocity distribution, and thus to account for droplet trajectory crossings. Both features have been assessed, i.e., a high-order method in size and velocity, (Kah, Laurent, Massot, Fox, and Jay 2008). These methods have been implemented in the multi-fluid solver (MUSES3D) provided in this work. Nevertheless since they are not directly studied in the present work, we do not give their complete derivation. The precise description and evaluation of these methods will be provided in (Kah 2010).

The derivation of the multi-fluid model based on the monokinetic velocity distribution assumption, is detailed in this section. The steps for the derivation of this model from the kinetic level of description, detailed in Sec. 1.2, are provided. As we already mentioned, we decided to assess the efficiency of the multi-fluid model mainly through numerical studies, comparing it with a reference Lagrangian description, in the framework of simple droplet models, in order to conduct precise evaluation. We thus neglect collisions, coalescence and secondary breakup in the derivation of the multi-fluid system of conservation laws. We refer to (Massot and Villedieu 2001; Laurent, Massot, and Villedieu 2004; Dufour 2005), for the derivation of the multi-fluid model including coalescence and secondary breakup and to (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009) for rebound.

2.3.1 Semi-kinetic system conservation laws

The general form of the semi-kinetic system has been given in Eq. (2.19). We highlight here the main assumptions and give its form under an equilibrium assumption at zero temperature for the velocity. As previously mentioned, we use the surface S as the size variable, (Laurent and Massot 2001). In this framework, the closure of the system is obtained through the following assumptions :

[H1] For a given droplet size, at a given point $(t, \boldsymbol{x})$, there is only one characteristic averaged velocity $\boldsymbol{u}_d(t, \boldsymbol{x}, S)$ and one averaged temperature $T_d(t, \boldsymbol{x}, S)$.

[H2] The dispersion in the distribution function around the mean velocity $\boldsymbol{u}_d$ and temperature T_d is zero in each direction, whatever the point $(t, \boldsymbol{x}, S)$.

It is equivalent to presume the following NDF conditioned by droplet size :

$$f(t, \boldsymbol{x}, S, \boldsymbol{u}, T) = n(t, \boldsymbol{x}, S)\, \delta(\boldsymbol{u} - \boldsymbol{u}_d(t, \boldsymbol{x}, S))\, \delta(T - T_d(t, \boldsymbol{x}, S)), \tag{2.54}$$

that is to reduce the support of the NDF to a one dimensional sub-manifold parametrized by droplet size. The semi kinetic system is then close, and Eq. (2.19) becomes:

$$\begin{aligned} \partial_t\, n + \partial_{\boldsymbol{x}} \cdot (n\boldsymbol{u}_d) &= \partial_S(nR_{sd}), \\ \partial_t\, (n\boldsymbol{u}_d) + \partial_{\boldsymbol{x}} \cdot (n\boldsymbol{u}_d \otimes \boldsymbol{u}_d) &= \partial_S(nR_{sd}\boldsymbol{u}_d) + n\boldsymbol{F}_d, \\ \partial_t\, nh_{ld} + \partial_{\boldsymbol{x}} \cdot (nh_{ld}\boldsymbol{u}_d) &= \partial_S(nh_{ld}R_{sd}) + nC_{p,l}E_d, \end{aligned} \tag{2.55}$$

where

$$R_{sd} = R_s(\boldsymbol{u}_d, T_d), \quad h_{ld} = h_l(\boldsymbol{u}_d, T_d), \quad E_d = E(\boldsymbol{u}_d, T_d). \tag{2.56}$$

The expressions for the terms R_s and E are given in Sec. 1.2.2. It has to be noticed that the two descriptions at the "kinetic" level and at the moment level are equivalent as long as two conditions are satisfied. First the NDF conditioned by droplet size is mono-kinetic, i.e. satisfies Eq. (2.54) and associates one single velocity and one single temperature at a fixed position and time for a given droplet size. Second, the solution of the system of conservation equations remains smooth, see Chpt. 3 where non-regular $\delta-$shock solutions are studied.

2.3.2 Multi-fluid system conservation laws

Size discretization and closure

In order to obtain the classical multi-fluid system of conservation laws, we then discretize the phase space. We choose a discretization $0 = S^{(1)} < S^{(2)} < \cdots < S_p < \cdots < S^{(N_S+1)}$ for the droplet size phase space and to average the obtained system of conservation laws over each fixed size intervals $[S_p, S_{p+1}[$, called section. The set of droplets in one section can be seen as a "fluid" for which conservation equations are written. The sections exchange mass, momentum and heat. The average in each section consists in taking one moment in size, of order 3/2 in surface. The system of conservation laws gives thus the evolution of the vector of moments of the phase space $(S, \boldsymbol{u}, T)$:

$$\begin{pmatrix} M_{\frac{3}{2}\,\mathbf{0}\,0} \\ M_{\frac{3}{2}\,\mathbf{1}\,0} \\ M_{\frac{3}{2}\,\mathbf{0}\,1} \end{pmatrix}. \tag{2.57}$$

The order 3/2 in size $M_{\frac{3}{2}\,\mathbf{0}\,0}$ corresponds to the mass density of droplets and it has been chosen because of its relevance to the evaporation and combustion processes. The evolution of these moments gives the evolution of mass, momentum and enthalpy of the spray. To close the system, the following assumptions are introduced :

[H3] In one section, the characteristic averaged velocity and temperature do not depend on the size of the droplets.

[H4] The form of n as a function of S is supposed to be independent of t and $\boldsymbol{x}$ in a given section, thus decoupling the evolution of the mass concentration of droplets in a section from the repartition in terms of sizes.

These assumptions are equivalent to presume the NDF in velocity, temperature and size inside each section :

$$\begin{aligned} \boldsymbol{u}_d(t,\boldsymbol{x},S) &= \boldsymbol{u}_d^{(p)}(t,\boldsymbol{x}) && \forall S \in [S_p, S_{p+1}[, \\ T_d(t,\boldsymbol{x},S) &= T_d^{(p)}(t,\boldsymbol{x}) && \forall S \in [S_p, S_{p+1}[, \\ n(t,\boldsymbol{x},S) &= m^{(p)}(t,\boldsymbol{x})\,\kappa_S^{(p)}(S) && \forall S \in [S_p, S_{p+1}[, \end{aligned} \tag{2.58}$$

where $m^{(p)}$ is the mass concentration of droplets in the p^{th} section, given by:

$$m^{(p)}(t,\boldsymbol{x}) = \int_{S_p}^{S_{p+1}} \rho_l \frac{S^{3/2}}{6\sqrt{\pi}}\, n(t,\boldsymbol{x},S)\,\mathrm{d}S, \tag{2.59}$$

giving thus for the form $\kappa_S^{(p)}$ in one section:

$$\int_{S_p}^{S_{p+1}} \frac{\rho_l}{6\sqrt{\pi}} S^{3/2}\, \kappa_S^{(p)}(S)\, \mathrm{d}S = 1. \tag{2.60}$$

We have chosen, as advised in (Laurent and Massot 2001), a form for the NDF constant in radius, i.e., $\kappa_R^{(p)} = constant$. The formulation of the case $\kappa_S^{(p)} = constant$ is given in the last paragraph of Sec. 2.3.2. From Eq. (1.29), we have:

$$\kappa_S^{(p)}\, \mathrm{d}S = \kappa_R^{(p)}\, \mathrm{d}R, \tag{2.61}$$

leading to:

$$\kappa_S^{(p)}(S) = \frac{\kappa_R^{(p)}}{4\sqrt{\pi\, S}}. \tag{2.62}$$

We have then, through Eq. (2.60):

$$\kappa_R^{(p)} = \frac{48\pi}{\rho_l\,(S_{p+1}^2 - S_p^2)}. \tag{2.63}$$

System of conservation laws for the p^{th} section

We derive the evolution of the set of moments Eq. (2.57), from the semi-kinetic system Eq. (2.55):

$$\int_{S_p}^{S_{p+1}} \rho_l \frac{S^{3/2}}{6\sqrt{\pi}} \text{Eq. (2.55)}\, \mathrm{d}S, \tag{2.64}$$

and we obtain, under the closure assumption [H1] to [H4]:

$$\begin{aligned}
\partial_t\, m^{(p)} + \partial_{\boldsymbol{x}} \cdot \left(m^{(p)}\, \boldsymbol{u}_d^{(p)}\right) &= -\left(E_1^{(p)} + E_2^{(p)}\right) m^{(p)} + E_1^{(p+1)}\, m^{(p+1)}, \\
\partial_t\, \left(m^{(p)} \boldsymbol{u}_d^{(p)}\right) + \partial_{\boldsymbol{x}} \cdot \left(m^{(p)}\, \boldsymbol{u}_d^{(p)} \otimes \boldsymbol{u}_d^{(p)}\right) &= -\left(E_1^{(p)} + E_2^{(p)}\right) m^{(p)} \boldsymbol{u}_d^{(p)} + E_1^{(p+1)}\, m^{(p+1)} \boldsymbol{u}_d^{(p+1)} \\
&\quad + m^{(p)}\, \boldsymbol{F}_d^{(p)}, \\
\partial_t \left(m^{(p)}\, h_{l_d}^{(p)}\right) + \partial_{\boldsymbol{x}} \cdot \left(m^{(p)}\, h_{l_d}^{(p)}\, \boldsymbol{u}_d^{(p)}\right) &= -\left(E_1^{(p)} + E_2^{(p)}\right) m^{(p)} h_{l_d}^{(p)}\, E_1^{(p+1)}\, m^{(p+1)} h_{l_d}^{(p+1)} \\
&\quad + m^{(p)}\, C_{p,l}\, E_d^{(p)},
\end{aligned} \tag{2.65}$$

where we defined the averaged velocity and enthalpy in the p^{th} section:

$$\begin{aligned}
\boldsymbol{u}_d^{(p)}(t, \boldsymbol{x}) &= \frac{1}{m^{(p)}} \int_{S_p}^{S_{p+1}} \rho_l \frac{S^{3/2}}{6\sqrt{\pi}}\, \boldsymbol{u}_d(t, \boldsymbol{x}, S)\, n, \mathrm{d}S, \\
h_{l_d}^{(p)}(t, \boldsymbol{x}) &= \frac{1}{m^{(p)}} \int_{S_p}^{S_{p+1}} \rho_l \frac{S^{3/2}}{6\sqrt{\pi}}\, h_{ld}(t, \boldsymbol{x}, S)\, n, \mathrm{d}S,
\end{aligned} \tag{2.66}$$

External forces and heat exchange terms Given the size integration, the external forces and heat exchanges terms are given by:

$$\begin{aligned} \boldsymbol{F}_d^{(p)}(t,\boldsymbol{x}) &= \frac{1}{m^{(p)}} \int_{S_p}^{S_{p+1}} \rho_l \, \frac{S^{3/2}}{6\sqrt{\pi}} \, \boldsymbol{F}_d(t,\boldsymbol{x},S,\boldsymbol{u}_d) \, n(t,\boldsymbol{x},S) \, \mathrm{d}S, \\ E_d^{(p)}(t,\boldsymbol{x}) &= \frac{1}{m^{(p)}} \int_{S_p}^{S_{p+1}} \rho_l \, \frac{S^{3/2}}{6\sqrt{\pi}} \, E_d(t,\boldsymbol{x},S,\boldsymbol{u}_d,T_d) \, n(t,\boldsymbol{x},S) \, \mathrm{d}S. \end{aligned} \tag{2.67}$$

One has to notice that, for the physical droplet models used, the dependance of $\boldsymbol{F}_d$ and E_d in the variable size S can be assumed to be $1/S$, see Sec. 1.3. Furthermore, using the form for n, given in Eq. (2.58), and for $\kappa^{(p)}$, given in Eq. (2.63), the expressions for the averaged external force and rate of heat exchange become:

$$\begin{aligned} \boldsymbol{F}_d^{(p)}(t,\boldsymbol{x}) &= \boldsymbol{F}_d\left(t,\boldsymbol{x},S_d^{(p)},\boldsymbol{u}_d^{(p)}\right), \\ E_d^{(p)}(t,\boldsymbol{x}) &= E_d\left(t,\boldsymbol{x},S_d^{(p)},\boldsymbol{u}_d^{(p)},T_d^{(p)}\right), \end{aligned} \tag{2.68}$$

the mean surface for the p^{th} section, $S_d^{(p)}$, being given, for a radius constant reconstruction, Eq. (2.63):

$$S_d^{(p)} = \frac{S_p + S_{p+1}}{2}. \tag{2.69}$$

Evaporation terms Regarding the evaporation term, $\int_{S_p}^{S_{p+1}} \rho_l \, S^{3/2}/(6\sqrt{\pi}) \, R_{sd} \, n \, \mathrm{d}S$, an integration by part is conducted to separate the exchange terms between successive sections, given by:

$$E_1^{(p)} = \rho_l \frac{S_p^{3/2}}{6\sqrt{\pi}} \, R_{sd}^{(p)} \, \kappa_S^{(p)}, \tag{2.70}$$

from the exchange terms with the gaseous phase:

$$E_2^{(p)} = \int_{S_p}^{S_{p+1}} \rho_l \, \frac{\mathrm{d}_S(S^{3/2})}{6\sqrt{\pi}} \, R_{sd} \, \kappa_S^{(p)}(S) \, \mathrm{d}S, \tag{2.71}$$

that is, using Eq. (2.62) and Eq. (2.63)

$$\begin{aligned} E_1^{(p)} &= \frac{2\,S_p}{\left(S_{p+1}^2 - S_{p+1}^2\right)} \, R_{sd}^{(p)}, \\ E_2^{(p)} &= \frac{3}{(S_p + S_{p+1})} \, R_{sd}^{(p)}, \end{aligned} \tag{2.72}$$

with $R_{sd}^{(p)}$, given by:

$$R_{sd}^{(p)}(t,\boldsymbol{x}) = R_{sd}(t,\boldsymbol{x},S_d^{(p)},\boldsymbol{u}_d^{(p)},T_d^{(p)}). \tag{2.73}$$

These terms only depend on the choice of the size discretization on the choice of $\kappa^{(p)}$, they can thus be pre-computed, preliminarily to the system resolution. One can notice that to

conduct the integration of Eq. (2.71), we assumed that R_{sd} was not depending on surface inside the section. Actually, R_{sd} depends on size when we consider convective corrections for evaporation, as the correction depends on the relative Reynolds number $\mathrm{Re_p}$, given in Eq. (1.35). We thus assumed that this correction was changing slowly enough inside one section to consider it constant and take the point-wise value at the mean surface: $S = S_d^{(p)}$. The term $E_2^{(p)}$, allows to compute the fuel mass source term in the gaseous equation, $S_F^{species}$. We need to evaluate this term in order to study gaseous fuel mass fraction prediction by the multi-fluid model through the FADO procedure and for the coldflame computations, described in Sec. 1.4.3. For mono-component droplets, we have, Eq. (1.33)

$$S_F^{species} = S^m = \int_S \int_{\boldsymbol{u}} \int_T \rho_l \, R_s \, \frac{\mathrm{d}_S(S^{3/2})}{6\sqrt{\pi}} \, f \, \mathrm{d}S \, \mathrm{d}\boldsymbol{u} \, \mathrm{d}T, \tag{2.74}$$

and thus in the multi-fluid framework, the mass source term is given by the sum of the section contributions:

$$S_F^{species} = \sum_{p=1}^{N_S} m^{(p)} \int_{S_p}^{S_{p+1}} \rho_l \, \frac{\mathrm{d}_S(S^{3/2})}{6\sqrt{\pi}} \, R_{sd} \, \kappa_S^{(p)}(S) \, \mathrm{d}S \; = \sum_{p=1}^{N_S} m^{(p)} \, E_2^{(p)}. \tag{2.75}$$

Dimensionless multi-fluid formulation

The multi-fluid model is computed in a dimensionless form, obtained from the dimensionless kinetic formulation introduced in Sec. 1.3.4. The formulation uses, as in the previous pragraph, presumed shape constant in radius in a section, $\kappa_R^{(p)} = constant$. The non dimensional formulation with a shape constant in the surface variable, $\kappa_S^{(p)} = constant$, is provided in the next paragraph. We first introduce a reference mass density for droplets, m_0:

$$m_0 = \rho_\infty \frac{S_0^{3/2}}{6\sqrt{\pi}} n_0. \tag{2.76}$$

The dimensionless multi-fluid system of conservation laws is given by:

$$\begin{aligned} &\frac{x_0}{U_0 m_0} && \text{Eq. (2.65a)}, \\ &\frac{x_0}{U_0^2 m_0} && \text{Eq. (2.65b)}, \\ &\frac{x_0}{U_0 m_0 C_{p,\infty} T_\infty} && \text{Eq. (2.65c)}. \end{aligned} \tag{2.77}$$

It leads to the same system as Eq. (2.65), with the following dimensionless variable:

$$\begin{aligned}
\overline{m^{(p)}} &= \frac{m^{(p)}}{m_0}, \\
\overline{\boldsymbol{u}_d^{(p)}} &= \frac{\boldsymbol{u}_d^{(p)}}{u_0}, \\
\overline{h_{l_d}^{(p)}} &= \frac{h_{l_d}^{(p)}}{C_{p,\infty} T_\infty}.
\end{aligned} \tag{2.78}$$

The dimensionless form of the external forces, the rate of heat exchange and the surface rate of change in one section, are directly obtained from Sec. 1.3.4:

$$\begin{aligned}
\overline{\boldsymbol{F}_d^{(p)}}(\overline{t}, \overline{\boldsymbol{x}}) &= \overline{\boldsymbol{F}_d}\left(\overline{t}, \overline{\boldsymbol{x}}, \overline{S_d^{(p)}}, \overline{\boldsymbol{u}_d^{(p)}}\right), \\
\overline{E_d^{(p)}}(\overline{t}, \overline{\boldsymbol{x}}) &= \overline{E_d}\left(\overline{t}, \overline{\boldsymbol{x}}, \overline{S_d^{(p)}}, \overline{\boldsymbol{u}_d^{(p)}}, \overline{T_d^{(p)}}\right), \\
\overline{R_{s_d}^{(p)}}(\overline{t}, \overline{\boldsymbol{x}}) &= \overline{R_{sd}}\left(\overline{t}, \overline{\boldsymbol{x}}, \overline{S_d^{(p)}}, \overline{\boldsymbol{u}_d^{(p)}}, \overline{T_d^{(p)}}\right),
\end{aligned} \tag{2.79}$$

the expression of the of $\overline{\boldsymbol{F}_d}$, $\overline{E_d}$, and $\overline{E_d}$, being given in Eq. (1.98). The dimensionless terms for evaporation are given by:

$$\begin{aligned}
\overline{E_1^{(p)}} &= \overline{\rho_l}\, \overline{R_{s_d}^{(p)}}\, \overline{\kappa_R^{(p)}}\, \frac{1}{2} \overline{S_p}, \\
\overline{E_2^{(p)}} &= \overline{\rho_l}\, \overline{R_{s_d}^{(p)}}\, \overline{\kappa_R^{(p)}}\, \frac{3}{4} \left(\overline{S_{p+1}} - \overline{S_p}\right),
\end{aligned} \tag{2.80}$$

given

$$\overline{\kappa_R^{(p)}} = \frac{4}{\overline{\rho_l}\left(\overline{S_{p+1}}^2 - \overline{S_p}^2\right)}, \tag{2.81}$$

it leads:

$$\begin{aligned}
\overline{E_1^{(p)}} &= \frac{2\,\overline{S_p}}{\left(\overline{S_{p+1}}^2 - \overline{S_{p+1}}^2\right)} \overline{R_{s_d}^{(p)}}, \\
\overline{E_2^{(p)}} &= \frac{3}{\left(\overline{S_p} + \overline{S_{p+1}}\right)} \overline{R_{s_d}^{(p)}}.
\end{aligned} \tag{2.82}$$

The fuel mass source term becomes:

$$\overline{S_F^{species}} = \frac{\overline{S_0}^{3/2}}{6\sqrt{\pi}}\, N_0 \sum_{p=1}^{N_S} \overline{m^{(p)}}\, \overline{E_2^{(p)}}. \tag{2.83}$$

Surface constant reconstruction

The multi-fluid model formulation with a presumed NDF constant in surface within a size section is presented here. From Eq. (2.60), we have, assuming $\kappa_S^{(p)} = constant$:

$$\kappa_S^{(p)} = \frac{6\sqrt{\pi}}{\rho_l} \frac{5/2}{\left(S_{p+1}^{5/2} - S_p^{5/2}\right)}. \tag{2.84}$$

This new NDF form only modifies the source term expressions. As far as external forces and heat exchanges terms are concerned, the mean surface is modified. The p^{th} section mean surface $S_d^{(p)}$ for a surface constant reconstruction is given by:

$$S_d^{(p)} = \frac{3\left(S_{p+1}^{5/2} - S_p^{5/2}\right)}{5\left(S_{p+1}^{3/2} - S_p^{3/2}\right)}. \tag{2.85}$$

Concerning the evaporation source terms, $E_1^{(p)}$ and $E_2^{(p)}$ expressions can be computed from Eq. (2.70) and Eq. (2.71):

$$\begin{aligned} E_1^{(p)} &= \frac{5\, S_p^{3/2}}{2\left(S_{p+1}^{5/2} - S_p^{5/2}\right)} R_{sd}^{(p)}, \\ E_2^{(p)} &= \frac{5\left(S_{p+1}^{3/2} - S_p^{3/2}\right)}{2\left(S_{p+1}^{5/2} - S_p^{5/2}\right)} R_{sd}^{(p)}, \end{aligned} \tag{2.86}$$

In dimensionless formulation, $\kappa_S^{(p)}$ becomes:

$$\overline{\kappa_S^{(p)}} = \frac{5/2}{\overline{\rho_l}\left(\overline{S_{p+1}}^{5/2} - \overline{S_p}^{5/2}\right)}. \tag{2.87}$$

The variables $\overline{S_d^{(p)}}$, $\overline{E_1^{(p)}}$ and $\overline{E_2^{(p)}}$ are still given by Eq. (2.85), Eq. (2.86), but as functions of the dimensionless variables: $\overline{S_{p+1}}$, $\overline{S_p}$ and $\overline{R_{sd}^{(p)}}$.

Conclusion

This multi-fluid formulation is assessed throughout this work. A mathematical study of its governing system of equations, Eq. (2.65), is provided in Chpt. 3, leading to the design of an efficient numerical scheme, and to a theoretical characterization of the singularities arising due to the equilibrium assumption made for the velocity. The precision of this method is assessed through numerical, Chpt. 4, and experimental, Chpt. 5, comparisons in vortical flows. An original implementation and optimization of this method is provided in Part. III, allowing to tackle first 2-D jet and 3-D turbulent configurations, Part. IV.

Part II

Multi-fluid model mathematical, numerical and experimental characterization

Introduction

This part gives a first step toward the use of multi-fluid simulation in the more realistic configurations tackled in Part. IV. Indeed, the issues related to multi-dimensional multi-fluid simulation are tackled.
A three stage characterization, providing mathematical, numerical and experimental analysis of the multi-fluid method, is conducted in this part. It provides an extension of previous multi-fluid studies, as done in (Laurent and Massot 2001; Laurent, Massot, and Villedieu 2004; Dufour and Villedieu 2005; Laurent 2006), to multi-dimensional simulations. This part aims at precisely illustrating the robustness, the efficiency and the precision of multi-fluid multi-dimensional computations. The three levels considered allow to assess these issues.
The study of the multi-fluid system **mathematical** structure underlines the requirements to fulfill with the numerical scheme. Furthermore it gives the theoretical elements needed for a such scheme derivation. Finally, it allows to analyze the possible creation of singularities, due to droplet trajectory crossings, called $\delta-$shock. This study, conducted in Chpt. 3, leads to the derivation of a specific numerical scheme for multi-dimensional configurations. The robustness and efficiency of this numerical method is assesed in the **numerical** study conducted in Chpt. 4. The singularity analysis conducted in the mathematical part is shown to predict singularity occurence in the 2-D vortical flows considered. Finally, the precision of the method is assessed through quantitative comparisons with a DSMC Lagrangian method. The link between this Lagrangian approach and a Eulerian kinetic-based method, provided in Chpt. 2, shows the validity of such comparisons. In order to focus on the spray method used, the simplified modeling framework introduced in Chpt. 1 is used.
Finally, the ability of the multi-fluid method to capture the physics of spray/vortices interactions, is analyzed through **experimental** comparisons in Chpt. 5.
The coupling between this three complementary levels, constitutes a novel approach allowing a better physical comprehension. Indeed, the mathematical elements are needed to design efficient and robust numerical schemes that allow to study the physics of the spray. Furthermore, each studies brings in new elements:

- a numerical scheme for multi-dimensional Eulerian description able to simulate critical situations with high level of concentration, vacuum or singularities, is provided;
- a precise characterization of spray singularities, as well as their conditions of occurence, is conducted;
- time-resolved, size-conditioned quantitative comparisons between Eulerian and Lagrangian methods are achieved;
- finally, size-conditioned comparisons between a Eulerian spray description and experimental data are provided.

Chapter 3

Mathematical structure and resolution scheme

Contents

In the context of the evaluation of the multi-fluid relevance for industrial applications, the first step is to show the ability to design a robust and efficient numerical scheme. This step requires to analyze the mathematical properties of the multi-fluid model conservation laws. Isolating physical space transport, the monokinetic multi-fluid modeling leads to a weakly-hyperbolic system of conservation laws, known as the pressureless gas system, (Bouchut 1994). The numerical scheme used for its discretization must verify the positivity of density and a maximum principle on velocity. Furthermore, the scheme must handle the singularities, called $\delta-$shocks, that can arise. The link with the mathematical studies of theses singularities is given in Sec. 3.1, and illustrations for 2-D vortical flows are provided. Criteria predicting the formation of singularities, are defined from the provided illustrations. They will be assessed in Chpt. 4.
As far as phase space transport is concerned, one has to solve an ODE system, where stiffness can occur. Indeed, for instance the response time of the droplets to the drag force can be very different for different droplet inertia, and thus for different multi-fluid sections. It is thus necessary to choose carefully the appropriate integrator.
The multi-fluid numerical scheme must satisfy all these constraints and provide as genericity as possible, so that it does not reduce to a given multi-fluid method or to a given modeling framework. Its derivation is given in Sec. 3.2. An efficient transport scheme is provided for 2-D, 2-D axisymmetric and 3-D configurations. An efficient integrator for stiff ODE is used.

3.1 Mathematical peculiarities of multi-fluid modeling

The assumptions formulated at the kinetic and at the macroscopic level while deriving the Eulerian model, lead to several mathematical difficulties. Indeed we obtain a peculiar system of conservation equations without any pressure term leading to singularities called $\delta-$shocks. These singularities occur when the mono-kinetic assumption of the multi-fluid model ceases to be valid and introduces some artificial velocity averaging. The key issue is thus to characterize the appearance of such singularities and to relate it to the physics of droplet dynamics. This will have two consequences, first we will be able to choose comparison configurations which are within the validity limit of the multi-fluid model; second, since the configurations of interest frequently involve flow location which can be a little beyond the validity limit and still can be accurately described by the multi-fluid model, we aim at developing a numerical method robust enough to cope with these singularities in case they do occur.

3.1.1 Mathematical properties of monokinetic multi-fluid transport

In the monokinetic multi-fluid framework chosen in this study, Chpt. 2, the transport term has a peculiar structure known as the pressureless gas system, (Bouchut 1994). Indeed, it is similar to an Euler gas dynamics system of equations, but without any pressure term in the momentum equation:

$$\begin{aligned} \partial_t m + \partial_{\boldsymbol{x}} \cdot (m\boldsymbol{u}_d) &= 0, \\ \partial_t (m\boldsymbol{u}_d) + \partial_{\boldsymbol{x}} \cdot (m\boldsymbol{u}_d \otimes \boldsymbol{u}_d) &= 0. \end{aligned} \tag{3.1}$$

This weakly hyperbolic system can be found for example in astrophysics, when describing the formation of large scale structures in the universe or in the modeling of sticky particles (Zel′dovich 1970); it has also been presented in (Druzhinin and Elghobashi 1999) for bubbles. The structure of this system comes from the monokinetic assumption for the spray velocity distribution done at the kinetic level, Chpt. 2. An interesting property in this framework is the mathematical equivalence between the kinetic and the fluid level of description for smooth solutions, Theorem 2.

Theorem 2 (Bouchut). *Let $T > 0$, $m(t,\boldsymbol{x}), \boldsymbol{u}(t,\boldsymbol{x}) \in C^1([0,T] \times \mathbb{R}^d)$ and define*

$$f(t,\boldsymbol{x},\boldsymbol{u}) = m(t,\boldsymbol{x})\delta(\boldsymbol{u} - \boldsymbol{u}_d(t,\boldsymbol{x})). \tag{3.2}$$

Then $(m, \boldsymbol{u}_d)$ solve Eq. (3.1) in $[0,T] \times \mathbb{R}^d$ if and only if

$$\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{u} f) = 0 \quad \text{in } [0,T] \times \mathbb{R}^d \times \mathbb{R}^d, \tag{3.3}$$

where d is the number of physical space dimensions. This equivalence is shown in (Bouchut 1994), and is an interesting property for the construction of a numerical scheme, as detailed in Sec. 3.2. Therefore the solutions of the macroscopic system Eq. (3.1) are the solution of the kinetic equation Eq. (3.3) as long as the distribution is monokinetic, Eq. (3.2). A key point is thus to know whether an initially monokinetic distribution stays monokinetic as times evolves. These issues are discussed in (Jabin 2002), where a kinetic equation with a Stokes drag force source term is considered:

$$\partial_t f + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{u} f) = -\partial_{\boldsymbol{u}} \cdot \left(\frac{\boldsymbol{U_g} - \boldsymbol{u}}{\mathrm{St}} f \right). \tag{3.4}$$

This work considers on open domain with Lipschitz boundary. In the framework of our study, it is not needed to consider domain boundary, for periodicity reasons. The whole space $\mathbb{R}^d$ is thus considered. It is shown that, under normalization conditions on the initial velocity field $\boldsymbol{u}_d^0$ and on the gas velocity field $\boldsymbol{U_g}$, f remains monokinetic, Theorem 3. These conditions are based on the Stokes number.

Theorem 3 (Jabin). *Assume $m_0 \in L^1(\mathbb{R}^d)$ and*

$$\begin{aligned} & m \in L^\infty\left([0,\infty], L^1(\mathbb{R}^d)\right), \\ & \boldsymbol{u}_d \in L^\infty\left([0,\infty], W^{1,\infty}(\mathbb{R}^d)\right), \\ & ||\boldsymbol{u}_d^0||_{W^{1,\infty}(\mathbb{R}^d)} \leq \frac{1}{2d\,St}, \\ & ||\boldsymbol{U_g}||_{L^\infty\left([0,\infty], W^{1,\infty}(\mathbb{R}^d)\right)} \leq \frac{1}{4d\,St}. \end{aligned} \tag{3.5}$$

Then there exist unique m and $\boldsymbol{u}_d$ defined on the time interval $[0, \infty]$, solutions to Eq. (3.1).

A precise framework is thus defined where the equivalence between the kinetic and the fluid level of description stands. If the provided conditions are not fulfilled, this equivalence is no longer valid and singularities may occur at the fluid level of description. These singularities, caused by the breaking of the monokinetic assumption, are called $\delta-$shock. It corresponds to a discontinuity in velocity which leads to Dirac delta function concentrations in density. In fact, for smooth solutions, the equation on the velocity itself is decoupled from the conservation of mass and takes the form of the Burger's classical equation:

$$\partial_t \boldsymbol{u}_d + \boldsymbol{u}_d \cdot \partial_x \boldsymbol{u}_d = 0. \tag{3.6}$$

A shock may then arise, leading to the concentration of density at its interface. The breaking of the monokinetic assumption is related to droplet crossings trajectory. As mentioned in introduction of this manuscript, in the infinite Knudsen limit taken in the applications, cluster of droplets are crossing each other, without colliding. It is associated, in a Eulerian description of the spray, to a bimodal distribution at the kinetic level, leading to two droplet velocities at the same space and time location. In the resolution of the whole problem, as formulated in Eq. (2.65), the drag term will decrease the occurrence of such crossings, attracting droplet velocity toward gas velocity. Nevertheless, for droplets inertial enough, influence of drag is weaker and crossings may still occur. As we already mentioned in the assumption [H1], Sec. 2.2.3, at a given point there is only one averaged velocity, preventing the description of droplet crossings. In the two-fluid model, also based on one averaged velocity, the same problem occurs and droplet crossings can neither be described. However, in the multi-fluid model, as presented in assumption [H1], we have one averaged velocity at a given size. Consequently, the polydispersion described in the multi-fluid model allows crossing of droplets provided they do not belong to the same size interval. Nevertheless, equally-sized droplet crossings are out of the limits of the multi-fluid model and can not be described.
This observation has two consequences. First one wants to be able to control droplet dynamics for a given gaseous flow. Indeed for turbulent flows or even for laminar flows with contra-rotative vortices or impinging jets, equally-sized droplet crossings may occur. One would like to be able, through a limitation on the Stokes number of the droplets, to foresee these crossing occurrence. To do so, Theorem 3 is used in 2-D vortical flows to illustrate the conditions leading to droplet trajectory crossings. This study is conducted in Sec. 3.1.2. Second, a robust numerical method must be developed to cope with the velocity discontinuities and density concentration arising if equally-sized droplet crossings do still occur at rare occasions in the flow.

Along with the previously described singularities, the system can lead to creation of vacuum. One can note that it is important to be able to cope efficiently with vacuum zones since they represent areas of the flow where no droplet is to be found and are commonly encountered in most applications. Furthermore, the positivity of density and the maximum principle on velocity are preserved. One can note that configurations challenging to compute may arise, even when no singularity occurs, for example due to the concentration of density in an area of the flow. This situation occurs for example when droplets are ejected from the core of vortices and accumulated in weak vorticity areas.

A specific numerical method is consequently needed, for the transport part of the multi-fluid method, requiring:

- to stand high gradients even up to the situation where all the mass density would be concentrated in one cell;
- to preserve the positivity of mass density;
- to reproduce a discrete maximum principle on the velocity;
- to deal with vacuum zones in the computational domain.

Furthermore, the best compromise between robustness, precision and computational cost is required. The development of such a numerical method is presented in Sec. 3.2.

3.1.2 Multi-fluid limit illustrations in 1-D and 2-D vortical flows

The peculiar structure of the multi-fluid system and the set of associated assumptions require a precise analysis of droplet dynamics description. The purpose of this section is to illustrate the theoretical tools introduced in Sec. 3.1.1, in 1-D to 2-D cases, to define the conditions for singularities occurrence. A critical Stokes number is introduced characterizing the multi-fluid limits. For droplets with Stokes number under this criteria, the monokinetic multi-fluid assumption is valid and no singularity is to occur. Theorem 3 is first illustrated in a typical one dimensional problem where the critical Stokes number is defined. This definition is then extended to two-dimensional problems in the context of Taylor-Green vortices for the gaseous flow. The extension to a more general framework for more complex flow field is then considered. Finally Lagrangian numerical applications are performed in these configurations, to study the validity of the theoretical criterion introduced.

1-D model problem and critical Stokes number

The purpose of the present section is to identify the critical point for the appearance of "δ-shocks" (Bouchut 1994; Bouchut, Jin, and Li 2003), that is, the eventual concentration up to infinity of the droplet mass density field related to the creation of a discontinuous velocity field. Such an event corresponds to the crossing of characteristic curves in the physical space (LeVeque 2002) and to the limit of the mono-kinetic character of the NDF at the kinetic level, i.e. the velocity distribution at given location becomes multi-valued. These characteristic curves are defined, for both the kinetic equation Eq. (1.30) and the system of conservation laws Eq. (2.55), in the case of non-evaporating droplets, by a set of ordinary differential equations (ODE) and initial conditions :

$$\left\{\begin{array}{rcl} \mathrm{d}_t X_p &=& U_p \\ \mathrm{d}_t U_p &=& \dfrac{U_g(t,X_p)-U_p}{\mathrm{St}} \end{array}\right. , \quad \left\{\begin{array}{rcl} X_p(0) &=& X_p^0 \\ U_p(0) &=& U_p^0 \end{array}\right. , \tag{3.7}$$

with $U_p^0 = u^0(X_p^0)$ since the initial distribution is mono-kinetic. It should be noticed that the non-linear coupling between the two fields is contained in the fact that the gaseous velocity field U_g is only sampled by the droplet trajectory at X_p. Thus, the characteristic curves are the integral curves of the vector field defined by Eq. (3.7), parametrized by the initial spatial coordinate so that we will adopt the notation $(X_p, U_p)^t(t, X_p^0, U_p^0)$. Under some standard conditions on the regularity of the field U_g, the characteristic curves exist

and are well-defined for all time and spatial initial conditions. However, as soon as some characteristic curves cross each other in the only spatial projection of the characteristic diagram (x, t), the distribution ceases to be mono-kinetic and the equivalence between the macroscopic and kinetic descriptions is not valid any more. In fact the characteristics never cross each other in the (x, u) phase space so that the distribution admits multiple velocities at a point where the spatial projection of the characteristics cross each-other. For the sake of simplicity the generic example of steady gaseous flow field is given by a spatial harmonic oscillation $U_g(x) = \sin(2\pi x)$, with periodic boundary condition on the spatial interval $[0, 1]$. The initial condition for the spray is a uniform zero velocity distribution $u^0 = 0$, as well as a constant density distribution $n^0 = 1$. The characteristic first crossing point can be shown to be at $x = 1/2$ since this is the maximum of the gas velocity derivative. It can be seen as the point of maximal strain. In order to characterize the limit we linearize the original system of ODEs Eq. (3.7) at $x = 1/2$ for which the eigenvalues of the associated matrix are real, if and only if $8\pi\,\mathrm{St} \leq 1$. The limiting value of the Stokes number is then $1/(8\pi) \approx 0.0398$, that corresponds to the condition provided by Theorem 3. Taking a look at Fig. 3.1 will provide the reader with the intuitive picture of two cases where there is or is not characteristic crossing. We have represented on the left the trajectories of the particles (that are also the characteristic curves, for smooth solutions, in this pressureless configuration) with the usual convention that the abscissa is the spatial coordinate and the ordinate, the time evolution. We consider thirty equi-distributed initial spatial positions and a zero initial velocity. We plotted the evolution of the position versus time for two Stokes numbers which are below and above the critical one. It can be clearly seen that the characteristics cross at $x = 0.5$ for $\mathrm{St} = 0.3$ at time around $t = 0.5$ which corresponds to the first time when the velocity distribution at $x = 0.5$ becomes multi-valued as shown in the position-velocity phase plane on the top-right of Fig. 3.1. On the contrary, such a scenario never occurs for $\mathrm{St} = 0.03$ for which the characteristic curves never cross and the velocity field as a function of position always remain mono-kinetic as presented in Fig. 3.1, (bottom).

Besides, as shown in (Dufour 2005) in the 1D case, the droplet velocity field is rapidly attracted, within a non dimensional time equal to a few Stokes number, to an invariant velocity manifold. It is smooth only if the non-dimensional Stokes number is below the critical limit and becomes discontinuous beyond this threshold, thus allowing the droplets to go from one half of the domain to the other (see Fig. 3.1, (top), for $\mathrm{St} = 0.3$). This manifold is easily observed in the (X_p, U_p) phase plane in Fig. 3.1, (top-right), for $\mathrm{St} = 0.03 < 1/(8\pi)$.

For Stokes numbers beyond the critical limit, let us underline the fact that, even for the Williams equation at the kinetic level, there is a singularity at the time when the characteristics are crossing in the (x, t) diagram. At this exact time, the zeroth order moment of the NDF, that is the number density of droplets, becomes infinite at $x = 1/2$ and the original modeling at the kinetic level can cease to be valid if the initial droplet number density is high enough for the collision term to become important in the neighborhood of the axis of symmetry where the singularity occurs. Even if this singularity is spatially integrable, the original modeling on the NDF should then involve a collision term or a "granular pressure". For an interesting study of the influence of the initial number density of droplets on the collisional term in the NDF equation in a different framework, we refer to the work of Volkov and collaborators, see (Volkov, Tsirkunov, and Oesterle 2005) and references therein.

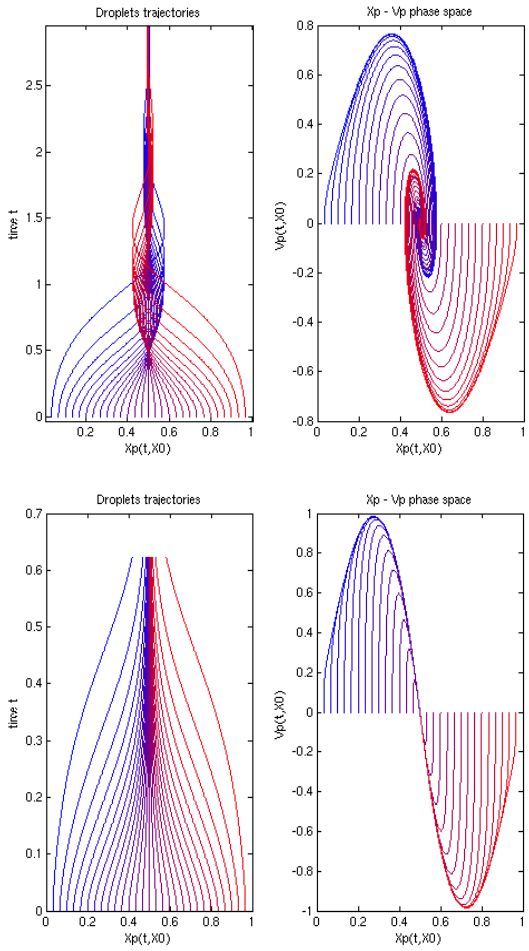


Figure 3.1 – *(top): $St = 0.3$, characteristic crossing in the (x, t) plane (left) and phase plane dynamics (x, u) (right) for 30 initial conditions equally distributed in space with zero initial velocity; (bottom): $St = 0.03$, no characteristic crossing in the (x, t) plane (left) and phase plane dynamics (x, u) (right) for 30 initial conditions equally distributed in space with zero initial velocity.*

This will prove to be symptomatic of what happens in multi-dimensional configurations with more complex flows.

Taylor-Green vortices

In a first step toward more complex multi-dimensional configurations, we investigate a gaseous flow field which is given by the two-dimensional Taylor Green vortices, a steady solution of the inviscid incompressible Euler equations. The reason for such a choice is related to the fact that while being two-dimensional, such a configuration is representative of the vortical structure of turbulent flows and still allows an analytical treatment that will highlights the study of more complex configurations. To extend the study to a two-dimensional version of the steady spatially oscillating gaseous flow field, we consider a steady solution of the incompressible Euler equations with periodic boundary conditions, which reads in the non-dimensional setting $U_g = \sin(2\,\pi\,x)\,\cos(2\,\pi\,y)$ for the horizontal

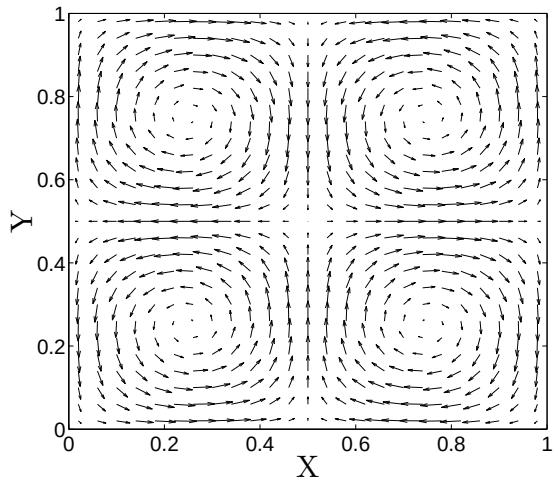


Figure 3.2 – *Velocity vectors of Taylor-Green gaseous vortices configuration.*

velocity, and $V_g = -\cos(2\,\pi\,x)\,\sin(2\,\pi\,y)$ for the vertical one, $(x, y) \in [0, 1] \times [0, 1]$. The structure of the flow field is presented in Fig. 3.2 through the velocity vectors.

In order to analytically characterize the critical Stokes number, we focus on the behavior of the system around the central point $(1/2, 1/2)$. The characteristics, in their non-dimensional form, are then linearized at this point and it yields :

$$\begin{cases} \mathrm{d}_t X_p = U_p, \quad \mathrm{d}_t U_p = \dfrac{2\,\pi\,X_p - U_p}{\mathrm{St}}, \\ \\ \mathrm{d}_t Y_p = V_p, \quad \mathrm{d}_t V_p = \dfrac{-2\,\pi\,Y_p - V_p}{\mathrm{St}}. \end{cases} \tag{3.8}$$

The system of four ODEs can then be split into two decoupled system of ODEs in each direction which can be treated separately. In the x direction, the eigenvalues are always real, whereas, in the y direction, we recover the same analysis as in the 1D case, with the same critical value of the Stokes number. It can be shown that, for the considered initial mono-kinetic velocity distribution, the first point of characteristics crossing is at the center: $(1/2, 1/2)$ which is again the point of maximum rate of strain.

Beyond the obtained Stokes critical value, droplets are ejected from vortices and encounter droplets coming from other vortices, since the original number density of droplets is symmetrical. Consequently, for Stokes numbers below this critical value, we are sure that the multi-fluid assumptions are valid in the sense that the kinetic modeling and the fluid modeling provide identical descriptions. For Stokes numbers beyond this critical value, the multi-fluid model and the kinetic model provide diverging solutions and are not equivalent any more. The Eulerian semi-kinetic and multi-fluid models both lead to infinite density concentrations and discontinuous velocity fields. In fact, in the non-dimensional gas velocity variables, the maximal value of the strain for both 1D and 2D cases is 2π at the symmetry point.

To obtain a first numerical illustration of the previously introduced Stokes criteria, we perform Lagrangian simulations in this Taylor-Green configuration to study droplets dynamics evolution with their inertia. The analytical velocity field allows us to compute analytically

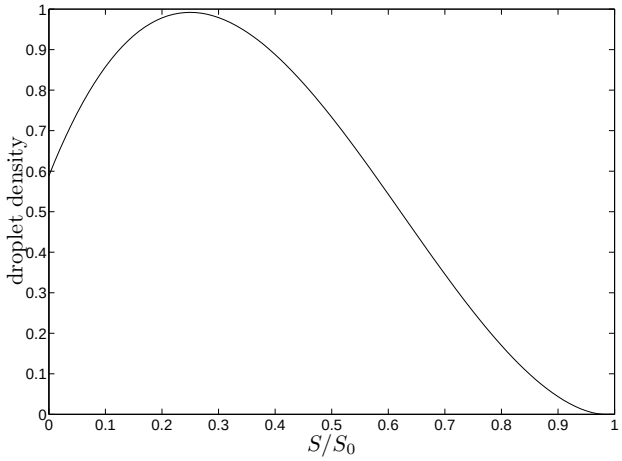


Figure 3.3 – *Initial conditions for droplets: non-dimensional polydisperse size distribution.*

the gas velocity at the position of the droplets. We decided to introduce in the computational domain a uniform distribution of droplets, equally distributed in the domain. Concerning the distribution in the size phase space, we have a polydisperse distribution reproducing $f^0(S)$ represented in Fig. 3.3. This droplet size distribution does not depend on spatial coordinate. We choose two maximum Stokes numbers for this distribution: a Stokes number under the critical value $\mathrm{St} = 0.03 < \mathrm{St_c}$ and one over the critical value $\mathrm{St} = 0.3 > \mathrm{St_c}$. Results are represented at four different times in Fig. 3.4 and Fig. 3.5. As expected, the droplets with a Stokes number under the critical value are ejected from the gaseous vortices, and they accumulate at the edges of the vortices without leaving their original vortex, see Fig. 3.4. On the contrary, more inertial droplets with a Stokes number over the critical value are ejected from their original vortex leading to crossing trajectories for droplets as shown in Fig. 3.5.

Extension to a general framework

The purpose of this subsection is to illustrate what happens in the framework of a more complex gaseous velocity field and to point out the similarities with what has just been presented in the context of Taylor-Green vortices.

In the previous study, we have considered only zero initial droplet velocity distribution and conducted a study of the critical Stokes number in terms of the gaseous flow field. In fact, in a general framework, there is also a condition on the initial droplet velocity field. From (Jabin 2002) it can be shown that system Eq. (2.19) is equivalent to the kinetic Williams equation for mono-kinetic initial velocity distributions under two conditions on both the initial velocity field $\boldsymbol{u}_d^0$ and on the maximum $a^{\max}$ of the derivative of the steady gaseous velocity field. The variable $a^{\max}$ denotes the maximal rate of strain of the gaseous flow field. In non-dimensional variables, the conditions can be written $a^{\max}\,\mathrm{St} < 1/(4\,d)$ and $|\partial_{\boldsymbol{x}}\overline{\boldsymbol{u}_d^0}|_\infty\,\mathrm{St} < 1/(2\,d)$, d being the number of dimension of the physical space. Since in the preceding case, $a^{\max} = 2\,\pi$, one recovers the obtained condition on the critical Stokes number in the one-dimensional setting. These two conditions insure, from a mathematical point of view, that the kinetic NDF will remain mono-kinetic, if it was originally so, for all

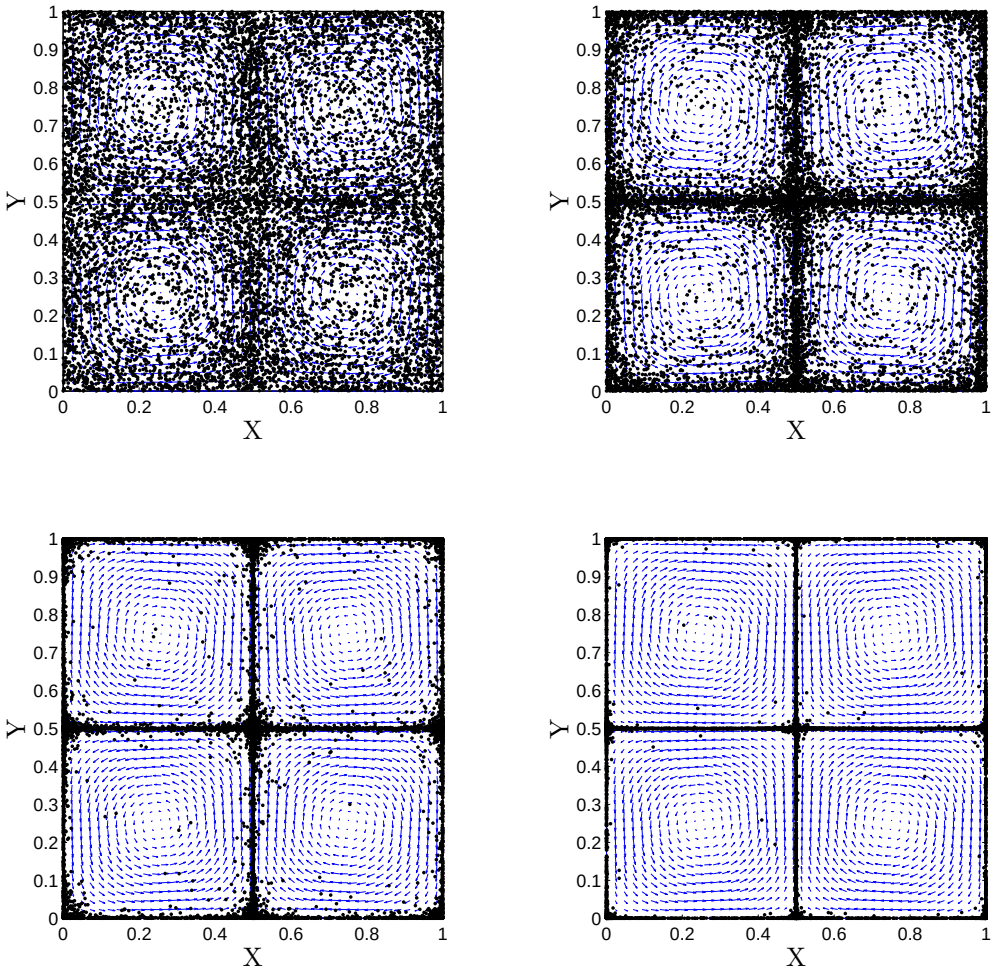


Figure 3.4 – *Lagrangian numerical parcels position at time $t = 0.5$, $t = 1.25$, $t = 2.75$, and $t = 4$ in the non dimensional setting. Positions are plotted over velocity vectors of the gaseous Taylor-Green vortices. The maximum Stokes number of the droplets distribution is $St = 0.03 < St_c$. Computation with 10000 numerical parcels.*

times. In this context this provides a rigorous basis to insure the validity of the semi-kinetic and multi-fluid model.

We can thus tackle the configuration of a given non-dimensional gaseous flow field. The point we want to make is related to the previous study, and relates to a new set of non-dimensional values of (x^0, a^0) for which a typical "eddy size" in the new space variable $x^+ = x/x^0$ and a related typical "rate of strain" in the new velocity variable $U_g^+ = U_g/(x^0\, a^0)$ are respectively one. In this framework, using the previous result, we will define the critical new Stokes number St_c^+ as being $1/4$, which will yield the original critical Stokes Number $\mathrm{St}_c = 1/(4\, a^0)$.

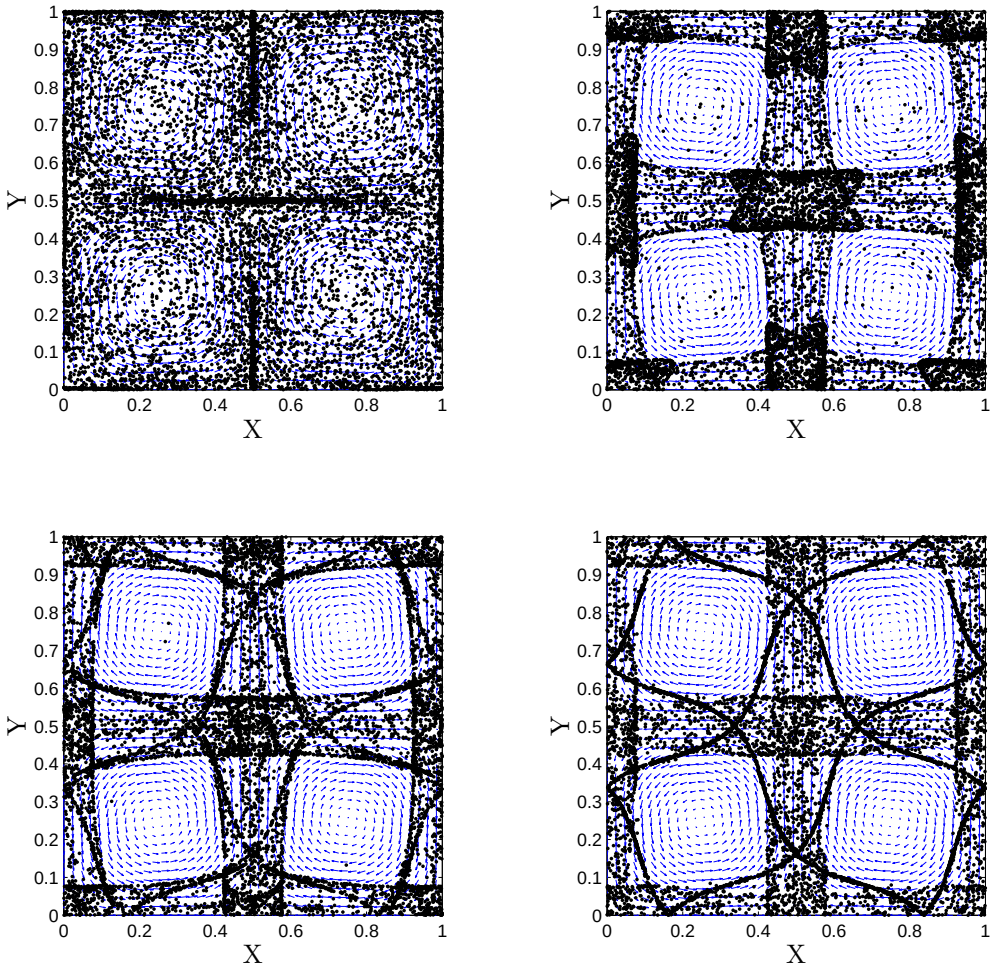


Figure 3.5 – *Lagrangian numerical parcels position at time $t = 0.5$, $t = 1.25$, $t = 2.75$, and $t = 4$ in the non dimensional setting. The positions are plotted over velocity vectors of the gaseous Taylor-Green vortices. The maximum Stokes number of the droplets distribution is $St = 0.3 > St_c$. Computation with 10000 numerical parcels.*

The only point remaining to be fulfilled is the choice of the couple (x^0, a^0). In the Taylor Green vortices, this choice is obvious, since the typical vortex size is $x^0 = 1$ and the typical rate of strain is $a^0 = 2\pi$, thus yielding the obtained value $\mathrm{St_c} = 1/8\pi$.

As long as the flow is incompressible, the two eigenvalues of the symmetric part of the velocity tensor are of opposite sign and the positive eigenvalue is then presented. More precisely :

$$\partial_{x,y} \begin{bmatrix} U_g \\ V_g \end{bmatrix} = \begin{bmatrix} \partial_x U_g & \dfrac{\partial_y U_g + \partial_x V_g}{2} \\ \dfrac{\partial_y U_g + \partial_x V_g}{2} & \partial_y V_g \end{bmatrix} + \frac{\omega}{2} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix}, \tag{3.9}$$

where $\omega = \partial_y U_g - \partial_x V_g$ is the vorticity field and $\pm i\omega/2$ are the two eigenvalues of the second matrix. The eigenvalues of the first matrix read :

$$\lambda_\pm = \frac{1}{2}\left(\partial_x U_g + \partial_y V_g \pm \sqrt{\psi(U_g, V_g)}\right), \tag{3.10}$$

with:

$$\psi(U_g, V_g) = (\partial_x U_g - \partial_y V_g)^2 + (\partial_y U_g + \partial_x V_g)^2 . \tag{3.11}$$

For the case of study of an incompressible velocity field in 2D, the zero-divergence of the flow imposes that there are two opposite eigenvalues. A characteristic value of a^0 can then be determined from the field of λ_+, from the distribution of this eigenvalue in the flow field. We can then determine the corresponding critical Stokes number: $\mathrm{St_c} = 1/(4\,a^0)$. Such definition will be used in Sec. 4.2, in the case of a turbulent gaseous flow field.

3.2 Eulerian multi-fluid specific numerical method

The numerical method designed for the multi-fluid method must deal with the complex mathematical structure of the governing system of equations, described and illustrated in Sec. 3.1. Furthermore, the numerical scheme has to present the best compromise between precision and cost, in order to tackle more complex 2-D jet or 3-D configurations, Part. IV. In order to treat efficiently the different difficulties of the multi-fluid system, that are a complex transport term and stiff source terms, we use an operator splitting, (Strang 1968; Bobylev and Ohwada 2001; Descombes and Massot 2004). It allows to design efficient numerical methods for each contribution. Furthermore, one takes advantage of the structured grids used in this study to introduce a dimensional splitting to treat multi-dimensional configurations, leading to a high efficiency. One has to note that this choice is not restrictive since extension to unstructured grids have been proposed in the literature. Methods devoted to unstructured grids would be required for industrial realizations. Nevertheless, albeit this work aims at studying the relevance of the multi-fluid method for industrial scale configurations, it focuses on a still academic context. This is a needed first step toward industrial applications, and with a clear gap in terms of configuration complexity compared to the multi-fluid computations provided up to now. Furthermore, this academic context is needed to provide precise analysis and validations of the model, Chpt. 4. Finally, its relative simplicity, allows to rapidly settle a configuration. It allows to realize different types of 2-D jets and to compute a first 3-D configuration, Part. IV.

3.2.1 Operator splitting

Phenomena involved in our problem are of two different types: transport induces an evolution in the physical space without leading to any interaction between the sections, whereas transport in internal coordinate space, i.e., size and velocity through evaporation, drag, heat transfer and collisions, induces an evolution without any coupling with the spatial coordinates. It is then interesting to separate them using an operator-splitting method. The multi-fluid system, Eq. (2.65), is then split into two systems, for the physical space

and for the phase space. On one hand, the evolution in physical space for the multi-fluid size section p reads:

$$\begin{aligned}
\partial_t\ m^{(p)} + \partial_{\boldsymbol{x}} \cdot \left(m^{(p)}\ \boldsymbol{u}_d^{(p)}\right) &= 0, \\
\partial_t\ \left(m^{(p)} \boldsymbol{u}_d^{(p)}\right) + \partial_{\boldsymbol{x}} \cdot \left(m^{(p)}\ \boldsymbol{u}_d^{(p)} \otimes \boldsymbol{u}_d^{(p)}\right) &= 0, \\
\partial_t \left(m^{(p)}\ h_{l_d}^{(p)}\right) + \partial_{\boldsymbol{x}} \cdot \left(m^{(p)}\ h_{l_d}^{(p)}\ \boldsymbol{u}_d^{(p)}\right) &= 0.
\end{aligned} \tag{3.12}$$

On the other hand, the phase space evolution, in internal coordinate space, reads for section p:

$$\begin{aligned}
\mathrm{d}_t\ m^{(p)} &= -\left(E_1^{(p)} + E_2^{(p)}\right)\ m^{(p)} + E_1^{(p+1)}\, m^{(p+1)}, \\
\mathrm{d}_t\ \left(m^{(p)} \boldsymbol{u}_d^{(p)}\right) &= -\left(E_1^{(p)} + E_2^{(p)}\right)\ m^{(p)} \boldsymbol{u}_d^{(p)} + E_1^{(p+1)}\, m^{(p+1)} \boldsymbol{u}_d^{(p+1)} + m^{(p)}\ \boldsymbol{F}_d^{(p)}, \\
\mathrm{d}_t \left(m^{(p)}\ h_{l_d}^{(p)}\right) &= -\left(E_1^{(p)} + E_2^{(p)}\right)\ m^{(p)} h_{l_d}^{(p)} + E_1^{(p+1)}\, m^{(p+1)} h_{l_d}^{(p+1)} + m^{(p)}\ C_{p,l}\ E_d^{(p)}.
\end{aligned} \tag{3.13}$$

The operator-splitting method consists in solving alternatively these simpler problems Eq. (3.12) and Eq. (3.13) in order to approximate the solution of the full problem, Eq. (2.65). We choose a Strang splitting which is second order in time provided all the steps are second order in time, with the following structure, see (Descombes and Massot 2004):

- Phase space transport: solve Eq. (3.12) during $\Delta t/2$,
- Physical space transport: solve Eq. (3.13) during Δt,
- Phase space transport: solve Eq. (3.12) during $\Delta t/2$.

This approach has the great advantage to preserve the properties of the schemes we use for the different contributions, as for example maximum principle or positivity. If we assume that the involved phenomena evolve at roughly similar time scales, this Strang splitting algorithm guaranties a second order precision in time provided that each of the elementary schemes has at least a second order time accuracy.

As mentioned in (LeVeque 2002), this Strang splitting can also be obtained with an algorithm based on a Lie splitting. The Lie splitting is composed of two steps of length Δt. The Strang splitting is constructed with this two steps but it alternates the order in which they are performed:

1. iteration 2n
   - Phase space transport: solve Eq. (3.12) during Δt,
   - Physical space transport: solve Eq. (3.13) during Δt,
2. iteration 2n+1
   - Physical space transport: solve Eq. (3.13) during Δt,
   - Phase space transport: solve Eq. (3.12) during Δt.

This form is in fact a classical Strang splitting with a splitting timestep $2\Delta t$. Indeed it can be written:

- Phase space transport: solve Eq. (3.12) during Δt
- Physical space transport: solve Eq. (3.13) during $2\Delta t$
- Phase space transport: solve Eq. (3.12) during Δt

The error due to the splitting increase, but the steps are of length Δt. Therefore there is no CFL reduction in the physical space transport. Furthermore, to advance the solution of Δt we need to perform only two steps and not three, and it can lead to a benefit in the computational efficiency of the algorithm.

3.2.2 Physical space transport resolution

As mentioned in Sec. 3.1.1, the transport part of the multi-fluid system of equations has a peculiar structure. Indeed the system for the physical transport of the p^{th} multi-fluid section, Eq. (3.12), is similar to the pressureless gas system Eq. (3.1). In order to explain the choice made for the multi-fluid transport scheme, one first focuses on this pressureless gas system. The kinetic finite volume scheme derived in (Bouchut, Jin, and Li 2003) is presented in a 1-D case. Afterward its extension to multi-dimensional cases is discussed. 2-D illustrations are first used as a framework to define the multi-dimensional method as efficient as possible for the cases considered in this work, Chpt. 4, Chpt. 5 and Part. IV. This general presentation gives the basis to derive the numerical method used for the multi-fluid transport. This transport term resolution is presented regardless of the multi-fluid section discretization, as physical transport is local in size phase space.

Finite volume kinetic schemes for pressureless gas system

The general framework of pressureless gas system is used to introduce finite volume kinetic methods, in 1-D and multi-dimensional cases.

Conservative methods We refer to (LeVeque 1992; LeVeque 2002; Godlewski and Raviart 1992), as reference manuals on conservative methods for hyperbolic problems. To introduce finite volume methods, we choose the 1-D system of conservation laws for pressureless gas dynamics:

$$\begin{array}{rcl} \partial_t\, m + \partial_x(m u_d) & = & 0, \\ \partial_t(m u_d) + \partial_x(m u_d^2) & = & 0. \end{array} \tag{3.14}$$

This system can be written in the form:

$$\partial_t\, \boldsymbol{p} + \partial_x(\boldsymbol{g}(\boldsymbol{p})) = 0, \tag{3.15}$$

where

$$\boldsymbol{p} = \begin{pmatrix} m \\ m\, u_d \end{pmatrix}; \quad \boldsymbol{g}(\boldsymbol{p}) = u_d\, \boldsymbol{p}. \tag{3.16}$$

In order to cope with the discontinuous solutions arising from the non linear conservation law Eq. (3.15), a finite volume method is well-suited. Indeed, finite difference methods can

be expected to break down near discontinuities in the solution, where the partial differential equation form of the conservation law, Eq. (3.15) does not hold in the classical sense, but in the sense of distributions. Finite volume methods are based on the integral form of the conservation law, instead of the differential equation. The integral form of Eq. (3.15) is given, for any two points x_1 and x_2:

$$\mathrm{d}_t \int_{x_1}^{x_2} \boldsymbol{p}(x,t)\,\mathrm{d}x = \boldsymbol{g}(\boldsymbol{p}(x_1,t)) - \boldsymbol{g}(\boldsymbol{p}(x_2,t)). \tag{3.17}$$

In the finite volume methods, rather than point-wise approximations at grid points, the domain is broken into grid cells, and the cell average of $\boldsymbol{p}$ is approximated over each grid cell. The classical finite volume discretization gives evaluation at the center of the cells: $0 < x_1 < x_2 < ... < x_{N_x}$, if N cells are considered. Therefore a cell corresponds to $[x_{i-1/2}, x_{i+1/2}]$, where: $0 \leq x_{1/2} < ... < x_{i-1/2} < ... < x_{N_x+1/2}$. The i^{th} cell average $\boldsymbol{p}_i^n$ of $\boldsymbol{p}$ at time t^n, is defined by the integral of $\boldsymbol{p}$ divided by the length of the cell:

$$\boldsymbol{p}_i^n = \frac{1}{\Delta x_i} \int_{x_{i-1/2}}^{x_{i+1/2}} \boldsymbol{p}(x,t^n)\,\mathrm{d}x, \tag{3.18}$$

where $\Delta x_i = x_{i+1/2} - x_{i-1/2}$.
These values are modified by the flux through the edge of the grid cells:

$$\boldsymbol{p}_i^{n+1} = \boldsymbol{p}_i^n - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+1/2} - \boldsymbol{F}_{i-1/2}\right), \tag{3.19}$$

where $\boldsymbol{F}$ is called the numerical flux function. This form can be naturally obtained from the time integration, over $[t^n, t^{n+1}]$, of the integral form Eq. (3.17). Therefore the numerical flux function can be seen as an average flux through $x_{i-1/2}$ over the time interval, (LeVeque 1992):

$$\boldsymbol{F}_{i-1/2} \sim \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \boldsymbol{g}(\boldsymbol{p}(x_{i-1/2},t))\,\mathrm{d}t. \tag{3.20}$$

The primary problem in finite volume methods is to determine good numerical flux functions that approximate the correct fluxes reasonably well, based on the approximate cell averages, the only data available. Note that in order to obtain high order resolution methods, the cell averages are used to reconstruct a piecewise linear function, used to obtain a more precise evaluation of the numerical flux function. It gives a better accuracy for smooth function, nevertheless, it fails near discontinuities where oscillations are generated. Slope limiters are thus used to detect discontinuity and to avoid local extrema creation. In our case, two main slope limiters are used, that gives, for example of the mass, $m = \boldsymbol{p}(1)$, slope Dm_i^n (LeVeque 2002):

- the minmod slope, defined by:

$$Dm_i^n = \mathrm{minmod}\left(\frac{m_i^n - m_{i-1}^n}{\Delta x_i^c}, \frac{m_{i+1}^n - m_i^n}{\Delta x_{i+1}^c}\right), \tag{3.21}$$

where

$$\mathrm{minmod}(a,b) = \begin{cases} a & \text{if} \quad |a| < |b| \quad \text{and} \quad ab > 0, \\ b & \text{if} \quad |b| < |a| \quad \text{and} \quad ab > 0, \\ 0 & \text{if} \quad ab \leq 0, \end{cases} \tag{3.22}$$

- the Monotonized Central-difference limiter (MC limiter), first introduced in (van Leer 1977) defined by:

$$Dm_i^n = \text{minmod}\left(\left(\frac{m_{i+1}^n - m_{i-1}^n}{\Delta x_{i-1}^c + \Delta x_{i+1}^c}\right), 2\left(\frac{m_i^n - m_{i-1}^n}{\Delta x_i^c}\right), 2\left(\frac{m_{i+1}^n - m_i^n}{\Delta x_{i+1}^c}\right)\right). \tag{3.23}$$

Furthermore, we want a numerical method able to cope with the delta-shock formation and the emergence of vacuum, described in Sec. 3.1.2. We then want a method that guaranties a maximum principle on the velocity and the positivity of density. These requirements enforces us to use a method of at most second-order.

Bouchut *et al.* proposed, in (Bouchut, Jin, and Li 2003), a second-order kinetic scheme designed to solve the pressureless gas system, thus ensuring positivity of density and maximum principle on velocity. They propose first- and second-order kinetic schemes for 1-D and 2-D pressureless gas systems.

1-D kinetic scheme algorithm A 1-D finite volume scheme kinetic scheme is developed in (Bouchut, Jin, and Li 2003) for this system. It is based on the equivalence between a macroscopic and a microscopic level of description, see Theorem 2:

$$\partial_t f + u\,\partial_x f = 0 \iff \left\{ \begin{array}{rcl} \partial_t\, m + \partial_x(m u_d) & = & 0 \\ \partial_t(m u_d) + \partial_x(m u_d^2) & = & 0 \end{array} \right. , \tag{3.24}$$

with:

$$f(t,x,u) = m(t,x)\delta(u - u_d(t,x)). \tag{3.25}$$

The values of m and u_d are then recovered from f by the formula:

$$\begin{pmatrix} m \\ m\,u_d \end{pmatrix}(t,\boldsymbol{x}) = \int_{\mathbb{R}} \begin{pmatrix} 1 \\ u \end{pmatrix} f(t,x,u)\mathrm{d}u. \tag{3.26}$$

In order to obtain the kinetic conservative scheme for the density and the momentum, we thus integrate the equivalent kinetic equation Eq. (3.24) (left) on the control volume $\mathcal{C}_{i,j} = \left[x_{i-1/2}, x_{i+1/2}\right]$ for all the velocities $u \in \mathbb{R}$ and for time $t \in \mathcal{T}^n = [t^n, t^{n+1}]$, that is to say we take the moment of order zero and one in velocity of the kinetic equation:

$$\int_{\mathcal{C}_{i,j}} \int_{\mathbb{R}} \int_{\mathcal{T}^n} \begin{pmatrix} \partial_t f \\ u\,\partial_t f \end{pmatrix} \mathrm{d}t\,\mathrm{d}x\,\mathrm{d}u = \int_{\mathcal{C}_{i,j}} \int_{\mathbb{R}} \int_{\mathcal{T}^n} \begin{pmatrix} u\,\partial_x f \\ u^2\,\partial_x f \end{pmatrix} \mathrm{d}t\,\mathrm{d}x\,\mathrm{d}u. \tag{3.27}$$

Let us define:

$$\begin{pmatrix} m_i^n \\ q_i^n \end{pmatrix} = \frac{1}{\Delta x_i} \int_{\mathcal{C}_{i,j}} \begin{pmatrix} m\,(t^n,x) \\ q\,(t^n,x) \end{pmatrix} \mathrm{d}x, \tag{3.28}$$

with $q_i^n = m_i^n\,u_i^n$ and $q = m\,u_d$. With these definitions, the integration Eq. (3.27) of the kinetic equation lead to the following scheme:

$$\begin{pmatrix} m_i^{n+1} \\ q_i^{n+1} \end{pmatrix} = \begin{pmatrix} m_i^n \\ q_i^n \end{pmatrix} - \frac{\Delta t}{\Delta x_i}\left(\boldsymbol{F}_{i+1/2} - \boldsymbol{F}_{i-1/2}\right), \tag{3.29}$$

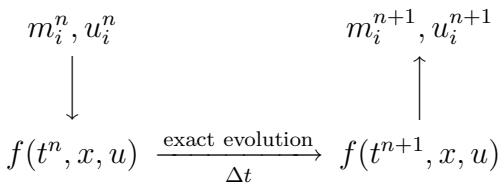


Figure 3.6 – *Main steps of the kinetic based transport scheme.*

with the fluxes given by:

$$\boldsymbol{F}_{i+1/2,j} = \begin{pmatrix} F^{(1)}_{i+1/2,j} \\ F^{(2)}_{i+1/2,j} \end{pmatrix} = \frac{1}{\Delta t} \int_{t^n}^{t^{n+1}} \int_{\mathbb{R}} \begin{pmatrix} 1 \\ u \end{pmatrix} u\, f(x_{i+1/2}, u, t) \mathrm{d}u\, \mathrm{d}t. \tag{3.30}$$

These fluxes are obtain under the CFL condition:

$$\Delta t \sup_x |u_d(x)| \leq \min_i \Delta x_i. \tag{3.31}$$

In order to compute these fluxes, an evaluation of the function f is needed at some points of the grid. This is done by reconstructing this function from the averaged values m_i^n and q_i^n. Let us denote $\tilde{f}$ this reconstructed function. This function, given by Eq. (3.25), can be taken as a piecewise constant function, for first-order scheme, Eq. (3.32) (left), or piecewise linear reconstruction, with adequate slope Dm_i^n, Du_i^n, Eq. (3.32) (right):

$$\begin{cases} m^n(x) = m_i^n \\ u^n(x) = u_i^n \end{cases}, \quad \begin{cases} m^n(x) = m_i^n + Dm_i^n(x - x_i) \\ u_d^n(x) = \bar{u}_i^n + Du_i^n(x - x_i) \end{cases}, \tag{3.32}$$

with

$$\bar{u}_i^n = u_i^n - \frac{Dm_i^n Du_i^n}{12 m_i^n} (\Delta x)^2. \tag{3.33}$$

The bar value $\bar{u}_i^n$ is introduced to ensure conservation of momentum, i.e., to satisfy Eq. (3.28). Finally, one needs to evaluate this reconstructed function, $\tilde{f}$, at time t^{n+1}. The kinetic equation Eq. (3.24) has an exact solution, a translation at velocity u_d. As $\tilde{f}$ is a solution of the kinetic equation, it can be written: $\tilde{f}(t, x, u) = \tilde{f}^n(x - u_d(t - t^n), u)$

The fluxes, Eq. (3.30), can now be computed, their form depend on the chosen reconstruction. Different reconstructions are described in (Bouchut, Jin, and Li 2003), from piecewise constant reconstruction to piecewise linear reconstruction, associated with a first and a second order precision, respectively. Besides, a simplified form of second order method, using constant reconstructions on half-cells and thus easier to extend to multi-dimensional configurations, is presented.

This kinetic scheme can be summarized as illustrated in Fig. 3.6, with the three steps allowing the scheme derivation and the flux computation: kinetic function reconstruction from average values, analytical kinetic evolution of the reconstruction, average values evolution, through numerical fluxes.

Multi-dimensional kinetic schemes formulation In order to discuss the extension of the previously described kinetic scheme, we first adopt a 2-D point of view. We analyze three types of second-order kinetic method extensions for multi-dimensional configurations. The analysis is based on 2-D examples:

- the method based on linear reconstruction can not be easily extended to 2-D configurations, the numerical fluxes computations being rather complex. This is all the more true for 3-D configurations. The expressions of the numerical fluxes for this method are not provided in (Bouchut, Jin, and Li 2003),
- the method using constant reconstructions on a grid cell sub-domains, half-cells in 1-D, and also providing second order accuracy, is thus better suited for multi-dimensional configurations. Such a method is described for 2-D configurations in (Bouchut, Jin, and Li 2003). In this case quarter-cell reconstructions are needed. For the example of the x-component velocity u, having $\boldsymbol{u} = (u, v)^t$, it leads:

$$\begin{aligned} u_{i,j}^{I} &= \bar{u}_{i,j} + \frac{\Delta x_i}{2} D^x u_{i,j}^n + \frac{\Delta y_j}{2} D^y u_{i,j}^n, \\ u_{i,j}^{II} &= \bar{u}_{i,j} - \frac{\Delta x_i}{2} D^x u_{i,j}^n + \frac{\Delta y_j}{2} D^y u_{i,j}^n, \\ u_{i,j}^{III} &= \bar{u}_{i,j} - \frac{\Delta x_i}{2} D^x u_{i,j}^n - \frac{\Delta y_j}{2} D^y u_{i,j}^n, \\ u_{i,j}^{IV} &= \bar{u}_{i,j} + \frac{\Delta x_i}{2} D^x u_{i,j}^n - \frac{\Delta y_j}{2} D^y u_{i,j}^n. \end{aligned} \tag{3.34}$$

  It corresponds to choose, for the half-cells constant values, the reconstruction taken at the corner of the cell. These reconstructions allow to compute the numerical fluxes. The resulting flux will depend on all the neighbors of the (i, j) cell. We can for example write for the first component, i.e., the mass density, in the first direction:

$$\begin{aligned} F_{i+1/2,j}^{(1)} = F(& m_{i,j}^n, m_{i-1,j}^n, m_{i-1,j-1}^n, m_{i-1,j+1}^n, m_{i,j-1}^n, m_{i,j+1}^n \\ & u_{i,j}^n, u_{i-1,j}^n, u_{i-1,j-1}^n, u_{i-1,j+1}^n, u_{i,j-1}^n, u_{i,j+1}^n). \end{aligned} \tag{3.35}$$

  These dependance on all the neighbor cells brings algorithmic complexity and makes the method more difficult to implement in complex industrial or semi-industrial codes. Furthermore it can brings higher computational cost, due to data memory access. Indeed, accessing to all the neighbors of a grid cell, may need to access to remote data in the processor memory. Finally, a half CFL condition is necessary when deriving this scheme, and it is difficult to find an efficient slope limiter not inducing important numerical diffusion.
- Another extension of constant sub-domain values can be proposed, where the constant values are computed at the center of the grid edges, instead of the corners. The x-

component velocity reconstruction Eq. (3.34), then becomes:

$$\begin{aligned} u_{i,j}^{I} &= \bar{u}_{i,j} + \frac{\Delta x_i}{2} D^x u_{i,j}^n, \\ u_{i,j}^{II} &= \bar{u}_{i,j} - \frac{\Delta y_j}{2} D^y u_{i,j}^n, \\ u_{i,j}^{III} &= \bar{u}_{i,j} - \frac{\Delta x_i}{2} D^x u_{i,j}^n, \\ u_{i,j}^{IV} &= \bar{u}_{i,j} + \frac{\Delta y_j}{2} D^y u_{i,j}^n. \end{aligned} \tag{3.36}$$

The corresponding numerical fluxes, valid under a half CFL condition, depend then only of one neighbor, $(i-1, j)$

$$F_{i+1/2,j}^{(1)} = F\left(m_{i,j}^n, m_{i-1,j}^n, u_{i,j}^n, u_{i-1,j}^n\right). \tag{3.37}$$

This method is easier to implement in complex industrial codes, may decrease the time needed for data memory access, and allows to take a MC slope limiter, see for example (LeVeque 1992), whereas the previous method is based on a more diffusive minmod slope limiter. Nevertheless it assumes that the only contribution to the flux is from the adjacent cell on the upwind side (the donor cell). Thus it fails to predict correctly a flow at an angle to the grid and can lead, in the Taylor-Green vortical flow previously described, to a high imprecision. Furthermore, this method has stability problems, as exposed in (LeVeque 2002). This type of methods is described as "Donor-Cell Upwind" (DCU) methods, and is shown, through a von Neumann stability analysis, to be stable only for Δt small enough that:

$$\left|\frac{u_d\ \Delta t}{\Delta x}\right| + \left|\frac{v_d\ \Delta t}{\Delta y}\right| \leq 1. \tag{3.38}$$

Given the difficulties linked to the multi-dimensional kinetic schemes, and since we aim at computing academic configurations with quadrilateral grids, the **dimensional splitting**, also called alternating direction technique (ADI), see (Godlewski and Raviart 1992) and (LeVeque 2002) and references therein, appears as a very interesting alternative. Indeed, it offers:

- an easy implementation,
- a great efficiency, since second order method with linear reconstruction can be used, as we perform a mono-dimensional transport,
- an easy extension to the aimed configurations, i.e. to 2-D, 2-D axisymmetric and 3-D domains.

One can note that this technique cannot be used for unstructured grids, nearly always encountered in industrial realizations. Therefore the study of a numerical scheme, with the greater efficiency in multi-dimensional configurations with unstructured mesh, is still to be done. For example, the detailed analysis of the numerical schemes implemented in the semi-industrial code AVBP, co-developed at IFP and CERFACS, provided in (Lamarque 2007), shows the necessity to develop a finite volume cell-vertex formulation of the kinetic scheme. Nevertheless, this is not the aim of the present work, where first realizations of the multi-fluid methods are given, providing ideas on the feasibility of such computations.

Numerical scheme for multi-fluid transport term

For the considered applications of this work, one wants to solve the multi-fluid system in 2-D and 3-D cases. The resolution of the transport part of this system is provided in this paragraph, using a kinetic scheme coupled with a dimensional splitting approach, taking advantage of the structured grids used.

We will use the same Strang splitting defined for the operator splitting, Sec. 3.2.1, in order to conserve the second order of the method. For a **2-D Cartesian** grid case, the multi-fluid transport term reads:

$$\begin{aligned}
\partial_t m + \partial_x \left(m u_d\right) + \partial_y \left(m v_d\right) &= 0, \\
\partial_t \left(m u_d\right) + \partial_x \left(m u_d^2\right) + \partial_y \left(m u_d v_d\right) &= 0, \\
\partial_t \left(m v_d\right) + \partial_x \left(m u_d v_d\right) + \partial_y \left(m v_d^2\right) &= 0, \\
\partial_t \left(m h_{l_d}\right) + \partial_x \left(m h_{l_d} u_d\right) + \partial_y \left(m h_{l_d} v_d\right) &= 0,
\end{aligned} \tag{3.39}$$

where u_d and v_d are the components of the mean multi-fluid velocity of the considered section. The index of the size section is dropped, as the transport is local in size phase space. To detail the splitting algorithm used, we introduce the operators, for the x- and y-direction:

$$\begin{aligned}
A_x &= \partial_t \cdot + \partial_x \left(u_d \cdot\right), \\
A_y &= \partial_t \cdot + \partial_y \left(v_d \cdot\right),
\end{aligned} \tag{3.40}$$

and the vector U:

$$\boldsymbol{U} = \begin{pmatrix} m \\ m u_d \\ m v_d \end{pmatrix}, \tag{3.41}$$

then the chosen Strang $2\Delta t$ timestep dimensional splitting algorithm can be written:

- obtain $\boldsymbol{U}^\star$ solving $\boldsymbol{A}_x \left(\boldsymbol{U}\right) = 0$ during Δt for x-direction,
- obtain $\boldsymbol{U}^{\star\star}$ solving $\boldsymbol{A}_y \left(\boldsymbol{U}^\star\right) = 0$ during $2\Delta t$ for y-direction,
- obtain $\boldsymbol{U}\left(t + 2\Delta t\right)$ solving $\boldsymbol{A}_x \left(\boldsymbol{U}^{\star\star}\right) = 0$ during Δt for x-direction.

As for the operator splitting, it is implemented with an alternation between the steps:

1. iteration 2n
   - obtain $\boldsymbol{U}^\star$ solving $\boldsymbol{A}_x \left(\boldsymbol{U}\right) = 0$ during Δt for x-direction,
   - obtain $\boldsymbol{U}\left(t + \Delta t\right)$ solving $\boldsymbol{A}_y \left(\boldsymbol{U}^\star\right) = 0$ during Δt for y-direction,
2. iteration 2n+1
   - obtain $\boldsymbol{U}^\star$ solving $\boldsymbol{A}_y \left(\boldsymbol{U}\right) = 0$ during Δt for y-direction,
   - obtain $\boldsymbol{U}\left(t + 2\Delta t\right)$ solving $\boldsymbol{A}_x \left(\boldsymbol{U}^\star\right) = 0$ during Δt for x-direction,

We have thus to solve, for each splitting step, a system that reads, for the example of the x-direction:

$$\begin{aligned}
\partial_t m + \partial_x (m u_d) &= 0, \\
\partial_t (m u_d) + \partial_x (m u_d^2) &= 0, \\
\partial_t (m v_d) + \partial_x (m v_d u_d) &= 0, \\
\partial_t (m h_{ld}) + \partial_x (m h_{ld} u_d) &= 0.
\end{aligned} \tag{3.42}$$

This system is solved with a 1-D cartesian kinetic scheme, based on the scheme described in (Bouchut, Jin, and Li 2003), and given in details in the following. One can see that the generic problem linked to Eq. (3.42), is the transport in the x-direction of a vector $\boldsymbol{V} = (m, m u_d, m\Psi_d)^t$. Indeed the transport equations of v_d and h_{ld} in the x-direction are the same. It corresponds to the advection of the scalar $m\Psi_d$, Ψ_d standing thus either for v_d or h_{ld}. Therefore, we only need to detail the 1-D kinetic scheme for the transport of the vector $\boldsymbol{V}$, to present the resolution of the 2-D Cartesian transport.

As far as **2-D axisymmetric** cases are concerned, one recovers exactly the same situation for the transport in the axial z-direction. The same scheme is thus used. Nevertheless, the transport in the radial direction has a different structure. Indeed, one needs to solve in the r-direction:

$$\begin{aligned}
\partial_t m + \frac{1}{r}\partial_r (r m u_d) &= 0, \\
\partial_t (m u_d) + \frac{1}{r}\partial_r \left(r m u_d^2\right) &= 0, \\
\partial_t (m v_d) + \frac{1}{r}\partial_r (r m u_d v_d) &= 0, \\
\partial_t (m h_{ld}) + \frac{1}{r}\partial_r (r m h_{ld} v_d) &= 0,
\end{aligned} \tag{3.43}$$

where u_d is the velocity in the radial direction and v_d in the axial one. The Cartesian 1-D scheme derived in (Bouchut, Jin, and Li 2003) has thus been extended in this work to radial 1-D configurations. We will thus describe the transport for the vector $\boldsymbol{V}$ in the 1-D radial case.

Finally, regarding **3-D** case, one recovers the transport term of the multi-fluid method, directly extended from the 2-D Cartesian case:

$$\begin{aligned}
\partial_t m + \partial_x (m u_d) + \partial_y (m v_d) + \partial_z (m w_d) &= 0, \\
\partial_t (m u_d) + \partial_x (m u_d^2) + \partial_y (m u_d v_d) + \partial_z (m u_d w_d) &= 0, \\
\partial_t (m v_d) + \partial_x (m v_d u_d) + \partial_y (m v_d^2) + \partial_z (m v_d w_d) &= 0, \\
\partial_t (m w_d) + \partial_x (m w_d u_d) + \partial_y (m w_d v_d) + \partial_z (m w_d^2) &= 0, \\
\partial_t (m h_{ld}) + \partial_x (m h_{ld} u_d) + \partial_y (m h_{ld} v_d) + \partial_z (m h_{ld} w_d) &= 0,
\end{aligned} \tag{3.44}$$

where u_d, v_d and w_d are the components of the mean multi-fluid velocity of the considered section. The 2-D cartesian operators, Eq. (3.40) are straightforwardly extended to obtain the following splitting algorithm:

- obtain $\boldsymbol{U}^{\star}$ solving $\boldsymbol{A}_x\left(\boldsymbol{U}\right)=0$ during Δt for x-direction,
- obtain $\boldsymbol{U}^{\star\star}$ solving $\boldsymbol{A}_y\left(\boldsymbol{U}^{\star}\right)=0$ during Δt for y-direction,
- obtain $\boldsymbol{U}^{3\star}$ solving $\boldsymbol{A}_z\left(\boldsymbol{U}^{\star\star}\right)=0$ during $2\Delta t$ for z-direction,
- obtain $\boldsymbol{U}^{4\star}$ solving $\boldsymbol{A}_y\left(\boldsymbol{U}^{3\star}\right)=0$ during Δt for y-direction,
- obtain $\boldsymbol{U}\left(t+2\Delta t\right)$ solving $\boldsymbol{A}_x\left(\boldsymbol{U}^{4\star}\right)$ during Δt for x-direction.

This algorithm is implemented, as for the 2-D Cartesian case, with an alternation of symmetrical steps. We have the to solve, for each splitting step, a system, that reads, for the example of the x-direction:

$$\begin{aligned}
\partial_t m+\partial_x\left(mu_d\right) &= 0,\\
\partial_t\left(mu_d\right)+\partial_x\left(mu_d^2\right) &= 0,\\
\partial_t\left(mv_d\right)+\partial_x\left(mv_du_d\right) &= 0,\\
\partial_t\left(mw_d\right)+\partial_x\left(mw_du_d\right) &= 0.\\
\partial_t\left(mh_{ld}\right)+\partial_x\left(mh_{ld}u_d\right) &= 0.
\end{aligned} \tag{3.45}$$

Therefore, the resolution corresponds, as in the 2-D Cartesian case, to the transport of the vector $\boldsymbol{V}=\left(m,mu_d,m\Psi_d\right)^t$, Ψ_d standing here for v_d, w_d or h_{ld}. The same 1-D Cartesian kinetic scheme is thus used.

We present in the following:

- a 1-D kinetic scheme for Cartesian direction, based on the work of (Bouchut, Jin, and Li 2003);
- a 1-D kinetic scheme for 1-D radial cases, derived in this work, extending (Bouchut, Jin, and Li 2003).

We choose to present the transport of the generic vector $\boldsymbol{V}=\left(m,mu_d,m\Psi_d\right)^t$. We thus choose the x-direction in the Cartesian case. The scheme is identical for y- or z-directions, taking $\boldsymbol{V}=\left(m,mv_d,m\Psi_d\right)^t$ and $\boldsymbol{V}=\left(m,mw_d,m\Psi_d\right)^t$, respectively.

1-D Cartesian kinetic scheme for multi-fluid transport The transport term solved in the multi-fluid framework, for x-direction:

$$\begin{aligned}
\partial_t m+\partial_x\left(mu_d\right) &= 0,\\
\partial_t\left(mu_d\right)+\partial_x\left(mu_d^2\right) &= 0,\\
\partial_t\left(m\Psi_d\right)+\partial_x\left(m\Psi_du_d\right) &= 0,
\end{aligned} \tag{3.46}$$

has the same structure as the pressureless gas system, Eq. (3.14). The 1-D kinetic scheme presented previously is used. The only difference comes from the additional equation on the scalar Ψ. We thus compute the following evolution:

$$\begin{pmatrix} m_i^{n+1}\\ m_i^{n+1}u_i^{n+1}\\ m_i^{n+1}\Psi_i^{n+1}\end{pmatrix}=\begin{pmatrix} m_i^{n}\\ m_i^{n}u_i^{n}\\ m_i^{n}\Psi_i^{n}\end{pmatrix}-\frac{\Delta t}{\Delta x_i}\left(\boldsymbol{F}_{i+1/2}-\boldsymbol{F}_{i-1/2}\right). \tag{3.47}$$

In order to compute the fluxes of the scheme, Eq. (3.30), we choose a linear piecewise reconstruction, ensuring a second order spatial precision. In the multi-fluid framework, Eq. (3.32) becomes:

$$\left\{\begin{array}{rcl} m^n(x) & = & m_i^n + Dm_i^n (x - x_i) \\ u_d^n(x) & = & \bar{u}_i^n + Du_i^n (x - x_i) \\ \Psi_d^n(x) & = & \bar{\Psi}_i^n + D\Psi_i^n (x - x_i) \end{array}\right. , \tag{3.48}$$

with

$$\bar{u}_i^n = u_i^n - \frac{Dm_i^n Du_i^n}{12 m_i^n}(\Delta x)^2, \quad \bar{\Psi}_i^n = \Psi_i^n - \frac{Dm_i^n D\Psi_i^n}{12 m_i^n}(\Delta x)^2. \tag{3.49}$$

The slope Dm_i^n is obtain here using a MC limiter to limit the numerical diffusion:

$$Dm_i^n = \left\{\begin{array}{ll} \min\left(2\frac{m_{i+1}^n - m_i^n}{\Delta x}, 2\frac{m_i^n - m_{i+1}^n}{\Delta x}, \frac{m_{i+1}^n - m_{i-1}^n}{2\Delta x}\right) & \text{if } m_{i-1}^n < m_i^n < m_{i+1}^n \\ \max\left(2\frac{m_{i+1}^n - m_i^n}{\Delta x}, 2\frac{m_i^n - m_{i-1}^n}{\Delta x}, \frac{m_{i-1}^n - m_{i-1}^n}{2\Delta x}\right) & \text{if } m_{i-1}^n > m_i^n > m_{i+1}^n \\ 0 & \text{otherwise} \end{array}\right. . \tag{3.50}$$

This is different to what is done in (Bouchut, Jin, and Li 2003) where a minmod limiter was used. The slopes Du_i^n and Dv_i^n are chosen in order to guarantee the maximum principle property on the velocity and also the CFL-like condition $\Delta t Du_i^n > -1$, see (Bouchut, Jin, and Li 2003):

$$\begin{array}{rcl} Du_i^n & = & \frac{1}{2}\left(\mathrm{sgn}(u_{i+1}^n - u_i^n) + \mathrm{sgn}(u_i^n - u_{i-1}^n)\right) \\ & & \times \min\left\{\frac{|u_{i+1}^n - u_i^n|}{(1 - \Delta x Dm_i^n/6m_i^n)\Delta x}, \frac{|u_i^n - u_{i-1}^n|}{(1 + \Delta x Dm_i^n/6m_i^n)\Delta x}, \frac{1}{\Delta t}\right\}, \end{array} \tag{3.51}$$

and

$$\begin{array}{rcl} D\Psi_i^n & = & \frac{1}{2}\left(\mathrm{sgn}(\Psi_{i-1}^n - \Psi_i^n) + \mathrm{sgn}(\Psi_i^n - \Psi_{i-1}^n)\right) \\ & & \times \min\left\{\frac{|\Psi_{i-1}^n - \Psi_i^n|}{(1 - \Delta x Dm_i^n/6m_i^n)\Delta x}, \frac{|\Psi_i^n - \Psi_{i-1}^n|}{(1 + \Delta x Dm_i^n/6m_i^n)\Delta x}, \frac{1}{\Delta t}\right\}. \end{array} \tag{3.52}$$

To give the fluxes at the interface $x_{i-1/2}$, let us denote $m_{i-1/2}^L$, $m_{i-1/2}^R$, $u_{i-1/2}^L$, $u_{i-1/2}^R$, $\Psi_{i-1/2}^L$, $\Psi_{i-1/2}^R$ the corresponding values of $m^n(x)$, $u_d^n(x)$ and $\Psi_d^n(x)$ at the left and the right of interface $x_{i-1/2}$ between the cells $(i-1)$ and (i):

$$\begin{array}{rclrcl} m_{i-1/2}^L & = & m_{i-1}^n + \frac{\Delta x}{2} Dm_{i-1}^n, & m_{i-1/2}^R & = & m_i^n - \frac{\Delta x}{2} Dm_i^n, \\ u_{i-1/2}^L & = & \bar{u}_{i-1}^n + \frac{\Delta x}{2} Du_{i-1}^n, & u_{i-1/2}^R & = & \bar{u}_i^n - \frac{\Delta x}{2} Du_i^n, \\ \Psi_{i-1/2}^L & = & \bar{\Psi}_{i-1}^n + \frac{\Delta x}{2} D\Psi_{i-1}^n, & \Psi_{i-1/2}^R & = & \bar{\Psi}_i^n - \frac{\Delta x}{2} D\Psi_i^n. \end{array} \tag{3.53}$$

Finally, the fluxes can be written in the vector flux splitting form, integrating separately over $\mathbb{R}^+$ and $\mathbb{R}^-$:

$$\begin{aligned} F^{(1)}_{i-1/2} &= F^{+(1)}_{i-1/2} + F^{-(1)}_{i-1/2}, \\ F^{(2)}_{i-1/2} &= F^{+(2)}_{i-1/2} + F^{-(2)}_{i-1/2}, \\ F^{(3)}_{i-1/2} &= F^{+(3)}_{i-1/2} + F^{-(3)}_{i-1/2}, \end{aligned} \tag{3.54}$$

so we can write:

$$\begin{aligned} F^{+(1)}_{i-1/2} &= m^L_{i-1/2}\left(\alpha^L_{i-1/2}\right)_+ - \frac{\Delta t}{2}\left(\alpha^L_{i-1/2}\right)^2_+ \Lambda^{+(1)}_{i-1/2}, \\ F^{-(1)}_{i-1/2} &= m^R_{i-1/2}\left(\alpha^R_{i-1/2}\right)_- - \frac{\Delta t}{2}\left(\alpha^R_{i-1/2}\right)^2_- \Lambda^{-(1)}_{i-1/2}, \\ F^{+(2)}_{i-1/2} &= m^L_{i-1/2}\left(\alpha^L_{i-1/2}\right)^2_+ - \frac{\Delta t}{2}\left(\alpha^L_{i-1/2}\right)^2_+ \Lambda^{+(2)}_{i-1/2}, \\ F^{-(2)}_{i-1/2} &= m^R_{i-1/2}\left(\alpha^R_{i-1/2}\right)^2_- - \frac{\Delta t}{2}\left(\alpha^R_{i-1/2}\right)^2_- \Lambda^{-(2)}_{i-1/2}, \\ F^{+(3)}_{i-1/2} &= m^L_{i-1/2}\Psi^L_{i-1/2}\left(\alpha^L_{i-1/2}\right)_+ - \frac{\Delta t}{2}\left(\alpha^L_{i-1/2}\right)^2_+ \Lambda^{+(3)}_{i-1/2}, \\ F^{-(3)}_{i-1/2} &= m^R_{i-1/2}\Psi^R_{i-1/2}\left(\alpha^R_{i-1/2}\right)_- - \frac{\Delta t}{2}\left(\alpha^R_{i-1/2}\right)^2_- \Lambda^{-(3)}_{i-1/2}, \end{aligned} \tag{3.55}$$

with the following definitions:

$$\begin{aligned} \alpha^L_{i-1/2} &= \frac{u^L_{i-1/2}}{1+\Delta t D u^n_{i-1}}, \\ \left(\alpha^L_{i-1/2}\right)_+ &= \frac{\left(u^L_{i-1/2}\right)_+}{1+\Delta t D u^n_{i-1}}, \qquad \left(u^L_{i-1/2}\right)_+ = \max\left(0, u^L_{i-1/2}\right), \end{aligned} \tag{3.56}$$

and

$$\boldsymbol{\Lambda^+}_{i-1/2} = \begin{pmatrix} \Lambda^{+(1)}_{i-1/2} \\ \Lambda^{+(2)}_{i-1/2} \\ \Lambda^{+(3)}_{i-1/2} \end{pmatrix} = \begin{pmatrix} Dm^n_{i-1} \\ -m^L_{i-1/2} D u^n_{i-1} + \alpha^L_{i-1/2} D m^n_{i-1} + \dfrac{\Delta t}{3} D m^n_{i-1} D u^n_{i-1} \alpha^L_{i-1/2} \\ m^L_{i-1/2} D\Psi^n_{i-1} + \Psi^L_{i-1/2} D m^n_{i-1} - \dfrac{2\Delta t}{3} D m^n_{i-1} D\Psi^n_{i-1} \alpha^L_{i-1/2} \end{pmatrix}. \tag{3.57}$$

These flux expressions can then be summarized as follow, for the $\mathbb{R}^+$ contribution:

$$\boldsymbol{F}^+_{i-1/2} = \boldsymbol{U}^L_{i-1/2}\left(\alpha^L_{i-1/2}\right)_+ - \frac{\Delta t}{2}\left(\alpha^L_{i-1/2}\right)^2_+ \boldsymbol{\Lambda^+}_{i-1/2}\ , \tag{3.58}$$

with:

$$\boldsymbol{U}^L_{i-1/2} = \begin{pmatrix} m^L_{i-1/2} \\ m^L_{i-1/2}\ \alpha^L_{i-1/2} \\ m^L_{i-1/2}\ \Psi^L_{i-1/2} \end{pmatrix}, \quad \boldsymbol{F}^+_{i-1/2} = \begin{pmatrix} {F^+}^{(1)}_{i-1/2} \\ {F^+}^{(2)}_{i-1/2} \\ {F^+}^{(3)}_{i-1/2} \end{pmatrix}. \tag{3.59}$$

1-D radial kinetic scheme for multi-fluid transport For the 1-D radial transport, the system takes the form:

$$\begin{aligned} \partial_t m + \frac{1}{r}\partial_r \left(r m u_d\right) &= 0, \\ \partial_t \left(m u_d\right) + \frac{1}{r}\partial_r \left(r m u_d^2\right) &= 0, \\ \partial_t \left(m \Psi_d\right) + \frac{1}{r}\partial_r \left(r m \Psi_d v_d\right) &= 0. \end{aligned} \tag{3.60}$$

The same techniques as in (Bouchut, Jin, and Li 2003) can then be used in order to derive a second order scheme which preserves the maximum principle on the velocities and the positivity of m. In this radial framework, the kinetic equation, equivalent to the system Eq. (3.60), takes the form:

$$\partial_t f + \frac{u}{r}\partial_r \left(r\, f\right) = 0, \quad (t, r, u, v) \in \mathbb{R}^+ \times \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}, \tag{3.61}$$

with :

$$f\left(t, r, u, \Psi\right) = m\left(t, r\right) \delta\left(u - u_d\left(t, r\right)\right) \delta\left(\Psi - \Psi_d\left(t, r\right)\right), \tag{3.62}$$

Ψ_d being the mean axial velocity v_d or the mean enthalpy h_{ld} based on the mean temperature T_d. The values of m, u_d and Ψ_d are then recovered from f by the formula :

$$\begin{pmatrix} m \\ m\, u_d \\ m\, \Psi_d \end{pmatrix} (t, r) = \int_{\mathbb{R}^2} \begin{pmatrix} 1 \\ u \\ \Psi \end{pmatrix} f\left(t, r, u, \Psi\right) \mathrm{d}u\mathrm{d}\Psi. \tag{3.63}$$

One defines the discretization $0 = r_{-1/2} < r_{1/2} < \ldots < r_{i+1/2} < \ldots$ over $\mathbb{R}^+$ and sets $\Delta r_i = r_{i+1/2} - r_{i-1/2}\ \forall\ i \in \{0, 1, \ldots\}$. The scheme is a finite volume method giving approximations m_i^n, u_i^n and v_i^n of averaged values on each cell $[r_{i-1/2}, r_{i+1/2}]$ of m, u and v at each discrete time t^n :

$$\begin{pmatrix} m_i^n \\ m_i^n u_i^n \\ m_i^n \Psi_i^n \end{pmatrix} \simeq \frac{1}{r_i \Delta r_i} \int_{r_{i-1/2}}^{r_{i+1/2}} r \begin{pmatrix} m(t^n, r) \\ m(t^n, r) u(t^n, r) \\ m(t^n, r) \Psi(t^n, r) \end{pmatrix} \mathrm{d}r. \tag{3.64}$$

The conservative scheme takes the form:

$$\begin{pmatrix} m_i^{n+1} \\ m_i^{n+1} u_i^{n+1} \\ m_i^{n+1} \Psi_i^{n+1} \end{pmatrix} = \begin{pmatrix} m_i^n \\ m_i^n u_i^n \\ m_i^n \Psi_i^n \end{pmatrix} - \frac{\Delta t}{r_i \Delta r_i} \left(\boldsymbol{F}_{i+1/2} - \boldsymbol{F}_{i-1/2}\right), \tag{3.65}$$

with the fluxes

$$\boldsymbol{F}_{i+1/2} = \frac{r_{i+1/2}}{\Delta t}\int_{t^n}^{t^{n+1}}\int_{\mathbb{R}^2}\begin{pmatrix}1\\u\\\Psi\end{pmatrix} uf\left(t, r_{i+1/2}, u\right)\mathrm{d}u\,\mathrm{d}\Psi\,\mathrm{d}t. \tag{3.66}$$

Note that $\boldsymbol{F}_{-1/2} = 0$.

As previously, we use the analytical kinetic solution $f\left(t, r, u, \Psi\right) = f^n\left(r - u\left(t - t^n\right), u, \Psi\right)$, and provide piecewise linear reconstruction at time $t = t^n$ and for r between $r_{i-1/2}$ and $r_{i+1/2}$:

$$\left\{\begin{array}{rcl} m^n\left(r\right) = & \bar{m}_i^n + Dm_i^n\left(r - r_i\right) \\ u_d^n\left(r\right) = & \bar{u}_i^n + Du_i^n\left(r - r_i\right) \\ \Psi_d^n\left(r\right) = & \bar{\Psi}_i^n + D\Psi_i^n\left(r - r_i\right) \end{array}\right. , \tag{3.67}$$

with

$$\begin{aligned} \bar{m}_i^n &= m_i^n - Dm_i^n\frac{\Delta r_i^2}{12r_i}, \\ \bar{u}_i^n &= u_i^n - \frac{\Delta r_i^2}{12r_i}Du_i^n\left[1 + \frac{Dm_i^n}{m_i^n}\left(r_i - \frac{\Delta r_i^2}{12r_i}\right)\right], \\ \bar{\Psi}_i^n &= \Psi_i^n - \frac{\Delta r_i^2}{12r_i}D\Psi_i^n\left[1 + \frac{Dm_i^n}{m_i^n}\left(r_i - \frac{\Delta r_i^2}{12r_i}\right)\right], \end{aligned} \tag{3.68}$$

in order to have the conservation property on the momentum, i.e., the property Eq. (3.64) is exact for the reconstructed functions. The slopes Dm_i^n, Du_i^n and $D\Psi_i^n$ are chosen, as for the Cartesian case, to satisfy the positivity of density and the maximum principle on velocity. We then choose, for $i > 0$:

$$\begin{aligned} Dm_i^n &= \frac{1}{2}\left[sgn\left(m_{i+1}^n - m_i^n\right) + sgn(m_i^n - m_{i-1}^n)\right] \\ &\quad \times \min\left\{\frac{\left|m_{i+1}^n - m_i^n\right|}{\Delta r_i\left(1 - \frac{\Delta r_i}{6r_i}\right)}, \frac{\left|m_i^n - m_{i-1}^n\right|}{\Delta r_i\left(1 + \frac{\Delta r_i}{6r_i}\right)}\right\}, \end{aligned} \tag{3.69}$$

$$\begin{aligned} Du_i^n &= \frac{1}{2}\left[sgn\left(u_{i+1}^n - u_i^n\right) + sgn\left(u_i^n - u_{i-1}^n\right)\right] \\ &\quad \times \min\left\{\frac{\left|u_{i+1}^n - u_i^n\right|}{\Delta r_i\left(1 - \beta_i\right)}, \frac{\left|u_i^n - u_{i-1}^n\right|}{\Delta r_i\left(1 + \beta_i\right)}\right\}, \end{aligned} \tag{3.70}$$

and

$$\begin{aligned} D\Psi_i^n &= \frac{1}{2}\left[sgn\left(\Psi_{i-1}^n - \Psi_i^n\right) + sgn\left(\Psi_i^n - \Psi_{i-1}^n\right)\right] \\ &\quad \times \min\left\{\frac{\left|\Psi_{i-1}^n - \Psi_i^n\right|}{\Delta r_i\left(1 - \beta_i\right)}, \frac{\left|\Psi_i^n - \Psi_{i-1}^n\right|}{\Delta r_i\left(1 + \beta_i\right)}\right\}, \end{aligned} \tag{3.71}$$

with

$$\beta_i = \frac{\Delta r_i}{6r_i}\left[1 + \frac{Dm_i^n}{m_i^n}\left(r_i - \frac{\Delta r_i}{12r_i}\right)\right]. \tag{3.72}$$

For $i = 0$, the axial symmetry at $r = 0$ imposes that $Dm_0^n = 0$, $D\Psi_0^n = 0$ and the formulas Eq. (3.70) can be used for Du_0^n if we define $u_{-1}^n = -u_0^n$. In order to write the flux at interface $r_{i-1/2}$, we introduce the value at the left and at the right of the interface, as done for the Cartesian case in Eq. (3.53). We can then write the flux in the vector flux splitting form defined in Eq. (3.54). In this radial case, the fluxes can be summarized in the form:

$$\boldsymbol{F}^+_{i-1/2} = \boldsymbol{U}^L_{i-1/2}\left(\alpha^L_{i-1/2}\right)_+ Ra^+_{i-1/2} - \frac{\Delta t}{2}\left(\alpha^L_{i-1/2}\right)^2_+ \left(\boldsymbol{\Lambda}^+{}_{i-1/2}\right)_{\text{axi}}, \tag{3.73}$$

with:

$$\left(\boldsymbol{\Lambda}^+{}_{i-1/2}\right)_{\text{axi}} = \begin{pmatrix} \lambda_1\ Rb^+_{i-1/2} \\ -\lambda_2\ Rb^+_{i-1/2} + \lambda_3\ Rc^+_{i-1/2} + \lambda_4\ Rd^+_{i-1/2} \\ \lambda_5\ Rb^+_{i-1/2} + \lambda_6\ Rc^+_{i-1/2} - \lambda_7\ Re^+_{i-1/2} \end{pmatrix}, \tag{3.74}$$

where the λ_i are the terms of vector $\boldsymbol{\Lambda}^+{}_{i-1/2}$, see Eq. (3.57):

$$\boldsymbol{\Lambda}^+{}_{i-1/2} = \begin{pmatrix} \lambda_1 \\ \lambda_2 + \lambda_3 + \lambda_4 \\ \lambda_5 + \lambda_6 - \lambda_7 \end{pmatrix}, \tag{3.75}$$

and with:

$$\begin{aligned} Ra^+_{i-1/2} &= r_{i-1/2} - \frac{\Delta t}{2}\left(\alpha^L_{i-1/2}\right)_+, \\ Rb^+_{i-1/2} &= r_{i-1/2} - \frac{\Delta t}{3}\left(\alpha^L_{i-1/2}\right)_+, \quad Rc^+_{i-1/2} = r_{i-1/2} - \frac{2\Delta t}{3}\left(\alpha^L_{i-1/2}\right)_+, \\ Rd^+_{i-1/2} &= r_{i-1/2} - \frac{2\Delta t}{4}\left(\alpha^L_{i-1/2}\right)_+, \quad Re^+_{i-1/2} = r_{i-1/2} - \frac{3\Delta t}{4}\left(\alpha^L_{i-1/2}\right)_+. \end{aligned} \tag{3.76}$$

We conducted different numerical tests to validate the numerical scheme we have derived. A Rieman problem at a point $r > 0$ has for example been tested inducing the development of a delta-shock that the numerical scheme capture with the correct propagation speed. Since it is not very different to what is done in (Bouchut, Jin, and Li 2003) (replacing m by rm), this test is not presented here. We merely focus on an other test case which emphasizes the particular properties of our system. Moreover, we presents only results for the m and u since, for Ψ, it is just a transport equation at the velocity u and it does not lead to any difficulty.

The computational domain is the one-dimensional segment [0,1] discretized with $0 = r_0 < r_1 < ... < r_{N_r} = 1$ such that $r_{i+1} - r_i = \Delta r$ for $i = 1, ..., N_r - 1$ and $r_1 - r_0 = \frac{\Delta r}{2}$. This ensures the symmetry around the axis $r = 0$. We take the initial data to be

$$m(0, r) = \begin{cases} 1/0.3 & \text{if } 0 \le r < 0.3 \\ 1/r & \text{if } 0.3 \le r < 1 \\ 0 & \text{if } 1 \le r \end{cases}, \tag{3.77}$$

and

$$u(0,r) = \begin{cases} -0.5 & \text{if } 0 \leq r < 0.35 \\ 0.4 & \text{if } 0.35 \leq r < 0.4 \\ 1.04 - 1.6r & \text{if } 0.4 \leq r < 0.9 \\ -0.4 & \text{if } 0.9 \leq r \end{cases} . \tag{3.78}$$

The exact solution is obtained using the characteristics for the variables $r\,m$ and u. At time $t = 0.5$, it is given by $r\,m(t,r) = \psi(t)\delta_0(r) + \mu(t,r)$, with $\psi(t) = \int_0^t r\,m(0,r)\mathrm{d}r = 5/48$, and

$$\mu(t,r) = \begin{cases} (0.25 + r)/0.3 & \text{if } 0 \leq r < 0.05 \\ 1 & \text{if } 0.05 \leq r < 0.1 \\ 0 & \text{if } 0.1 \leq r < 0.55 \\ 1 & \text{if } 0.55 \leq r < 0.6 \\ 5 & \text{if } 0.6 \leq r < 0.7 \\ 1 & \text{if } 0.7 \leq r < 0.8 \\ 0 & \text{if } 0.8 \leq r \end{cases} . \tag{3.79}$$

Concerning the velocity, the exact solution is drawn on Fig. 3.7, (bottom), and is given by:

$$u(t,r) = \begin{cases} -0.5 & \text{if } 0 \leq r < 0.1 \\ \text{undefined} & \text{if } 0.1 \leq r < 0.55 \\ 0.4 & \text{if } 0.55 \leq r < 0.6 \\ 0.4 - 8(r - 0.6) & \text{if } 0.6 \leq r < 0.7 \\ -0.4 & \text{if } 0.7 \leq r < 0.8 \\ \text{undefined} & \text{if } 0.8 \leq r \end{cases} . \tag{3.80}$$

In this test, the initial velocity jump to a higher value at $r = 0.35$, which leads to a vacuum state, followed by a linearly decreasing part, where the mass accumulates and causes the density to increase. Fig. 3.7 (bottom), and Fig. 3.7, (top-middle), are respectively the comparison of numerical and analytical solution of density and velocity. It can be seen that the scheme gives good results with, however, some numerical diffusion. Moreover, the initial negative velocity near $r = 0$ leads to an accumulation of mass density at $r = 0$, which is particular to our case representing the r part of a 2D axisymmetric problem. It is in fact a singularity for m but the averaged value is all the same defined, for the first cell :

$$m_1^n = \frac{2}{(r_1)^2}\left(\psi(t) + \int_0^{r_1} \frac{0.25 + r}{0.3}\mathrm{d}r\right). \tag{3.81}$$

For $\Delta r = 0.004$, the exact value of this quantity is 52133.48 whereas we obtain 52586 showing a good behavior of the scheme.

The expressions of the fluxes for the 1-D Cartesian transport, Eq. (3.58), and for the radial transport, Eq. (3.73), of the vector $\boldsymbol{V} = (m, u_d, \Psi_d)^t$ have thus been given. It allows to construct the dimensional splitting algorithm for 2-D, Cartesian or axisymmetric, and 3-D cases.

3.2.3 Phase space transport resolution

We need to solve, in the phase space transport step, the ODE system Eq. (3.13). The polydispersion resolution leads to a multi-scale problem. Indeed, droplets of different sizes

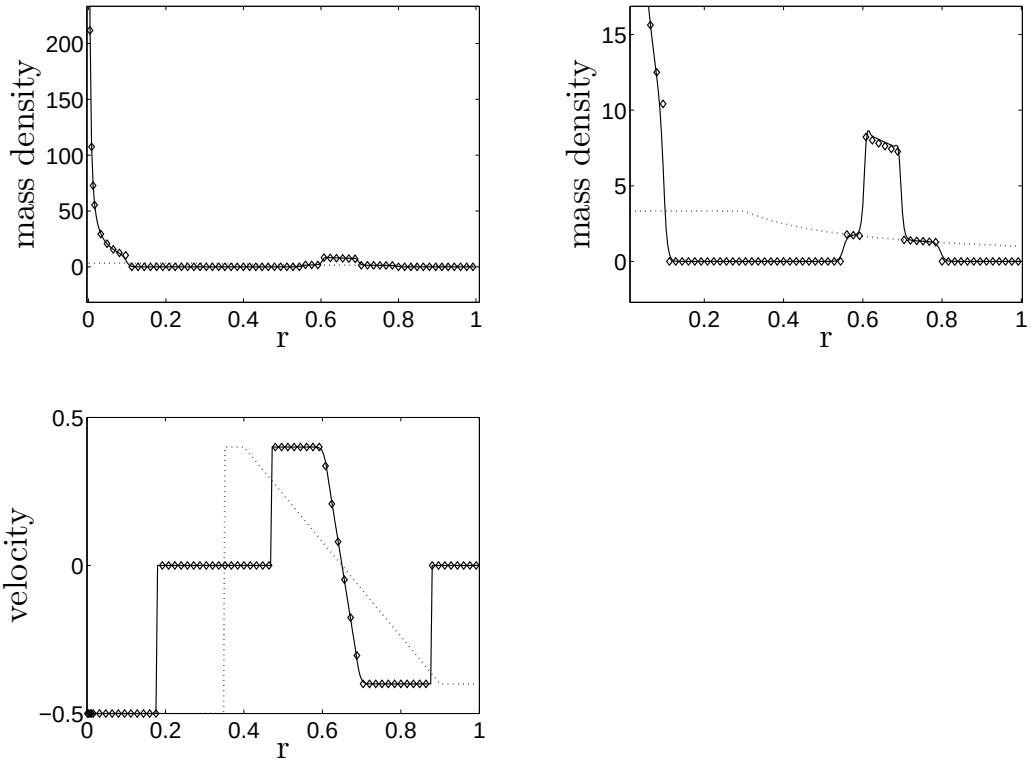


Figure 3.7 – *Comparison between the numerical (solid line), the analytical (diamonds) and the initial (dotted line) solution at time $t = 0.5$ with $CFL = 1$ and $\Delta r = 0.004$. (top-left) density fields with δ-shock formation at $r = 0$ (top-right) zoom on the density fields witout taking into account the behavior around $r = 0$ (bottom) velocity field.*

have different response time to the physical phenomena. The ODE integrator must then be carefully chosen.
Moreover, we reduce to simple model in some cases evaluating the resolution methods used. For example, only drag force and evaporation are taken into account in Chpt. 4 and Chpt. 5, dropping heat transfer, gravity and droplet interactions. We use, as a first approximation, a simple Stokes relaxation drag force and we consider a D^2 law for the evaporation process. In this context, it is interesting to treat separately the drag term and the evaporation term and to use a splitting algorithm, in order to obtain more efficient numerical methods.

Full ODE system resolution

In a more general context, we want to solve the full ODE system of the phase space transport, Eq. (3.13), with more complex models for evaporation and drag force, and possibly heat transfer and particle interactions, with an unsteady gas phase. We then want to use an ODE integrator for this system. This ODE resolution for the phase space transport is used in the computations presented in Part. IV, and in (de Chaisemartin, Freret, Kah,

Laurent, Fox, Reveillon, and Massot 2009) and (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009).

Integrator choice The evolution of the droplet behavior with their inertia can lead to several time scales in the problem, and then to stiffness. The Runge-Kutta implicit methods are particularly well suited for stiff problems, see for example (Hairer and Wanner 1996). If we want to find an approximation of the solution of:

$$\mathrm{d}_t(y) = \psi(t, y), \tag{3.82}$$

we can integrate Eq. (3.82), from t_0 to $t_0 + \Delta t$:

$$y\left(t_0 + \Delta t\right) = y_0 + \int_{t_0}^{t_0+\Delta t} \psi(t, y(t))\mathrm{d}t, \tag{3.83}$$

with $y_0 = y(t_0)$. To derive a numerical method, we have to approximate the integral, through a s-stage quadrature formula:

$$I = \int_{t_0}^{t_0+\Delta t} \psi(t, y(t))\mathrm{d}t \approx \Delta t \sum_{i=1}^{s} b_i \ \psi(t_0 + c_i, y(t_0 + c_i))\mathrm{d}t. \tag{3.84}$$

The coefficients c_i are the abscissa and b_i the weights of the quadrature formula. A first approximation could be to take a one step quadrature with $c = t_0 + \Delta t$ and $b = 1$:

$$I \approx \Delta t \psi(t_0 + \Delta t, y(t_0 + \Delta t)), \tag{3.85}$$

leading to the Euler implicit method:

$$y_1 = y_0 + \Delta t \ \psi(t_0 + \Delta t, y_1), \quad \text{with} \quad y_1 \approx y(t_0 + \Delta t). \tag{3.86}$$

To obtain a more precise method, one has to take more precise quadrature formula. We can, for example use a Radau 2-stages third order $(2s - 1)$ quadrature formula with $(b_1, b_2) = (3/4, 1/4)$ and $(c_1, c_2) = (1/3, 1)$, leading to the approximation:

$$y(t_0 + \Delta t) \approx y_0 + \frac{\Delta t}{4} \left(3 \ \psi\left(t_0 + \frac{\Delta t}{3}, y\left(t_0 + \frac{\Delta t}{3}\right)\right) + \psi\left(t_0 + \Delta t, y\left(t_0 + \Delta t\right)\right)\right). \tag{3.87}$$

We then need another quadrature to approximate the value $y\left(t_0 + \Delta t/3\right)$; we use a quadrature formula with the same abscissa as the quadrature formula Eq. (3.87):

$$y\left(t_0 + \frac{\Delta t}{3}\right) \approx y_0 + \frac{\Delta t}{12} \left(5 \ \psi\left(t_0 + \frac{\Delta t}{3}, y\left(t_0 + \frac{\Delta t}{3}\right)\right) - \psi\left(t_0 + \Delta t, y\left(t_0 + \Delta t\right)\right)\right). \tag{3.88}$$

We then have the scheme:

$$\begin{aligned} k_1 &= \psi\left(t_0 + \frac{\Delta t}{3}, y_0 + \frac{\Delta t}{12}\left(5k_1 - k_2\right)\right), \\ k_2 &= \psi\left(t_0 + \Delta t, y_0 + \frac{\Delta t}{4}\left(3k_1 + k_2\right)\right), \\ y_1 &= y_0 + \frac{\Delta t}{4}\left(3k_1 + k_2\right), \end{aligned} \tag{3.89}$$

defining a third-order implicit Runge Kutta method. More generally, a s-stage Runge Kutta method is defined for a_{ij}, b_i and c_i, for $i = 1, ..., s$ and $j = 1, .., s$, by:

$$\begin{aligned} k_i &= \psi\left(t_0 + c_i\Delta t, y_0 + \Delta t \sum_{j=1}^{s} a_{ij}k_j\right), i = 1, ..., s \\ y_1 &= y_0 + \Delta t \sum_{i=1}^{s} b_i k_i. \end{aligned} \tag{3.90}$$

When $a_{ij} = 0$ for $i \leq j$, we have an explicit Runge-Kutta method (ERK). If $a_{ij} = 0$ for $i < j$ and at least one $a_{ii} \neq 0$, we have a diagonal implicit Runge Kutta method (DIRK), or singly diagonal implicit method (SDIRK) if all diagonal elements are identical ($a_{ii} = \gamma$ for $i = 1, ..., s$. In all other cases, we speak of an implicit Runge Kutta method (IRK), (Hairer, Nørsett, and Wanner 1993).

The implicit Runge Kutta method based on Radau quadrature is called Radau IIA. We choose a 5 th order version of this integrator, obtained with a three-stage Radau quadrature of order $5 = 2s - 1$, still with $c_s = 1$. This method is shown to be A-stable in (Hairer and Wanner 1996). Furthermore, this method is L-stable, that allows a very fast convergence, with a fast high frequency absorption. This property is of importance for multi-fluid simulation as low inertia droplet velocity relaxes very rapidly to gas velocity, under drag force effect. An important limitation of this method, is the difficulty to obtain an efficient implementation of its algorithm.

Radau5 Program This Radau IIA method is implemented in the program RADAU5, written by Hairer, see (Hairer and Wanner 1996). It offers a very efficient algorithm, solving very efficiently the nonlinear system for the k_i, see Eq. (3.90). We will give here the main steps of this algorithm, refer to (Hairer and Wanner 1996) for details.

First of all, we rewrite the system Eq. (3.90), defining $g_i = y_0 + \sum_{j=1}^{s} a_{ij}k_j$, so that $k_i = \psi\left(t_0 + c_i\Delta t, g_i\right)$ and:

$$\begin{aligned} g_i &= y_0 + \Delta t \sum_{j=1}^{s} a_{ij}\psi\left(t_0 + c_j\Delta t, g_j\right), \quad i = 1, ..., s \\ y_1 &= y_0 + \Delta t \sum_{i=1}^{s} b_i \psi\left(t_0 + c_i\Delta t, g_i\right). \end{aligned} \tag{3.91}$$

In order to reduce the influence of round-off errors, we define the quantities:

$$z_i = g_i - y_0. \tag{3.92}$$

The scheme of Eq. (3.91) now reads:

$$\begin{aligned} z_i &= \Delta t \sum_{j=1}^{s} a_{ij}\psi\left(t_0 + c_j\Delta t, y_0 + z_j\right), \quad i = 1, ..., s \\ y_1 &= y_0 + \Delta t \sum_{i=1}^{s} b_i \psi\left(t_0 + c_i\Delta t, y_0 + z_i\right). \end{aligned} \tag{3.93}$$

The first step of the Radau5 program algorithm is the iterative resolution of the nonlinear system, given by the equation on the z_i in Eq. (3.93), with a Newton method. Basically, the Newton method can be summarized as follow. For the nonlinear system:

$$\psi(x) = 0, \tag{3.94}$$

where $\psi : \mathbb{R}^n \to \mathbb{R}^n$ is at least differentiable once. We start from a first approximation of the rough solution, x_0, and we linearize $\psi(x)$ around x_0:

$$\psi(x) \approx \psi(x_0) + \mathrm{d}_x\psi(x_0)(x - x_0), \tag{3.95}$$

and we find x verifying: $\psi(x_0) + \mathrm{d}_x\psi(x_0)(x - x_0) = 0$. The algorithm has then the form:

Algorithm 2 Newton method algorithm

while not(stopping criterion) **do**
  compute $\psi(x^k)$ and $\mathrm{d}_x\psi(x^k)$
  solve $\mathrm{d}_x\psi(x^k)\Delta x^k = -\psi(x^k)$
  compute $x^{k+1} = x^k + \Delta x^k$
end while

In our case, for complex models, the Jacobian matrix is not known analytically and is then evaluated numerically. We then use a simplified Newton method, where the Jacobian matrix is approximated by $\mathrm{d}_x\psi(x_0)$, so that we have:

$$\mathrm{d}_x\psi(x_0)\Delta x^k = -\psi(x^k). \tag{3.96}$$

Therefore the Jacobian matrix is only computed once during the iterative procedure. For the Radau IIA method, the nonlinear system Eq. (3.93) can be written:

$$z_i - \Delta t \ \textstyle\sum_{j=1}^{s} a_{ij}\psi\left(t_0 + c_j\Delta t, g_j\right) \ = \ 0, \quad i = 1, ..., s \tag{3.97}$$

thus the simplified Newton iteration read:

$$\begin{aligned} (\mathrm{Id} - \Delta t A \otimes J)\Delta Z^k &= -Z^k + \Delta t(A \otimes \mathrm{Id})F(Z^k), \\ Z^{k+1} &= Z^k + \Delta Z^k, \end{aligned} \tag{3.98}$$

with J, the approximated Jacobian:

$$J \approx \partial_y\psi\left(t_0, y_0\right), \tag{3.99}$$

and:

$$\begin{aligned} A &= (a_{ij}), \\ Z^k &= \left(z_i^k, ..., z_s^k\right)^t, \\ \Delta Z^k &= \left(\Delta z_i^k, ..., \Delta z_s^k\right)^t, \\ F(Z^k) &= \left(\psi(t_0 + c_1\Delta t, y_0 + z_i^k), , ..., \psi(t_0 + c_s\Delta t, y_0 + z_s^k)\right)^t. \end{aligned} \tag{3.100}$$

The details concerning the stopping criteria and the starting value for Newton iterations are not given here and can be found in (Hairer and Wanner 1996). It is then necessary to solve, in each iteration, the linear system given by the first equation of Eq. (3.98). A decomposition in two subsystems, leading to diminution of the number of operations is detailed in (Hairer and Wanner 1996) and implemented in the RADAU5 program.

This first step gives us the values of $(z_i), i = 1, ..., s$, we can then explicitly compute y_1, leading to s evaluations of ψ. This computation can be simplified when the matrix A is not singular. Indeed, in this case, Eq. (3.93) can be written:

$$A^{-1} \ (z_i, ..., z_s)^t \ = \ \Delta t \ \ \left(\psi\left(t_0 + c_1\Delta t, g_1\right), ...\psi\left(t_0 + c_s\Delta t, g_s\right)\right)^t, \tag{3.101}$$

then:

$$(b_1, ..., b_s)^t \ A^{-1} \ (z_i, ..., z_s)^t \ = \ \Delta t \sum_{i=1}^{s} b_i \ \psi\left(t_0 + c_i \Delta t, g_i\right). \tag{3.102}$$

Therefore, the evaluation of y_1, can be written:

$$y_1 \ = \ y_0 + \sum_{i=1}^{s} d_i z_i, \tag{3.103}$$

with

$$(d_1, ..., d_s) = (b_1, ..., b_s) \ \ A^{-1}. \tag{3.104}$$

In our 3-stage method, we have $d = (0, 0, 1)$, because $b_i = a_{si} \ \forall i$, and the method is stiffly accurate.

Finally, the method has an adaptative timestep, thus we need to define the step sizes. To do so, we use an embedded formula, that is to say we look for another evaluation $\widehat{y}_1$ of the solution using the same evaluations of ψ:

$$\widehat{y}_1 \ = \ y_0 + \Delta t \sum_{i=1}^{s} \widehat{b}_i k_i. \tag{3.105}$$

The error is estimated by the difference: $\widehat{y}_1 - y_1$. Since our method is of optimal order, we use a lower order method of order $\widehat{p}$ to find $\widehat{y}_1$. We can write the error estimation, see (Hairer and Wanner 1996):

$$y_1 - \widehat{y}_1 \approx C \ (\Delta t)^{\widehat{p}+1}. \tag{3.106}$$

The optimal Δt will correspond to an error estimation close to the tolerance:

$$Tol \approx C \ \Delta t_{opt}^{\widehat{p}+1}. \tag{3.107}$$

We can choose the new step size Δt_{new} equal to this optimal step size Δt_{opt}, given from Eq. (3.106) and Eq. (3.107):

$$\Delta t_{opt} = fac \ \Delta t \ \sqrt[\widehat{p}+1]{\frac{Tol}{||y_1 - \widehat{y}_1||}}, \tag{3.108}$$

where fac is a safety factor used to make the program safer. In the program RADAU5, this step evaluation is coupled with a predictive controller, useful for stiff problems with a rapid decrease of the step size. The idea of the previous method is to estimate error for step $n+1$, from data of the step n:

$$||err_{n+1}|| = C^n \Delta t^{\widehat{p}+1}. \tag{3.109}$$

This evaluation is based on the assumption $C^{n+1} \approx C^n$. A better model can be reached assuming that $\log C^n$ is a linear function of n, (Hairer and Wanner 1996). This means that $\log C^{n+1} - \log C^n$ is constant, or equivalently:

$$\frac{C^{n+1}}{C^n} \approx \frac{C^n}{C^{n-1}}. \tag{3.110}$$

It yields a new timestep evaluation, see (Hairer and Wanner 1996):

$$\Delta t_{new} = fac \ \Delta t^n \left(\frac{1}{||err_{n+1}||}\right)^{1/(\widehat{p}+1)} \frac{\Delta t^n}{\Delta t^{n-1}} \left(\frac{||err_n||}{||err_{n+1}||}\right)^{1/(\widehat{p}+1)}. \tag{3.111}$$

In the program RADAU5, the step size is taken as the minimum of the two step sizes Eq. (3.108) and Eq. (3.111).

Drag-evaporation simplified framework

We use, in this approximation, a modified Strang splitting and we will treat separately the two terms in a phase space transport step, alternating the order with iterations:

1. iteration 2n
   - solve evaporation term during Δt;
   - solve drag term during Δt;
2. iteration 2n+1
   - solve drag term during Δt;
   - solve evaporation term during Δt.

This framework is used for the multi-fluid evaluation through refined comparisons conducted in Chpt. 4.

Drag term We need to solve, for the drag term, the following system:

$$\left\{\begin{array}{rcl} \partial_t\ m^{(p)} & = & 0, \\ \\ \partial_t\ \left(m^{(p)}\boldsymbol{u}_d^{(p)}\right) & = & m^{(p)}\ \boldsymbol{F}_d^{(p)}, \end{array}\right. \tag{3.112}$$

with:

$$\boldsymbol{F}_d^{(p)} = \frac{1}{\mathrm{St}_d\ \tau_{\mathrm{conv}}}\left(\boldsymbol{U_g} - \boldsymbol{u}^{(p)}\right), \tag{3.113}$$

St_d being the Stokes number for the mean surface S_d: $\mathrm{St}_d = \mathrm{St}\,(S_d)$. This mean surface depends on the choice of the distribution's profile $\kappa^{(p)}$, as detailed in Chpt. 2. In the case of a steady gas phase, Chpt. 4, the system Eq. (3.112) can be solved exactly for a stationary gas velocity, insuring a good accuracy in time and leading to the scheme for the section p :

$$\begin{array}{rcl} m^{(p)\,n+1} & = & m^{(p)\,n}, \\ m^{(p)\,n+1}\boldsymbol{u}_d^{(p)\,n+1} & = & m^{(p)\,n}\left(\boldsymbol{U_g} + \left(\boldsymbol{u}_d^{(p)\,n} - \boldsymbol{U_g}\right)\exp\left(-\dfrac{\Delta t}{\mathrm{St}_d}\right)\right). \end{array} \tag{3.114}$$

The order of precision of the splitting step solving evaporation and drag force will thus depend on the order of the evaporation method.

Evaporation method To present the numerical method used to solve the evaporation term, we recall here the system of equations, for the p^{th} section, focusing only on this evaporation term:

$$\begin{array}{rcl} \mathrm{d}_t\ m^{(p)} & = & -\left(E_1^{(p)} + E_2^{(p)}\right)\ m^{(p)} + E_1^{(p+1)}\ m^{(p+1)}, \\ \mathrm{d}_t\ \left(m^{(p)}\boldsymbol{u}_d^{(p)}\right) & = & -\left(E_1^{(p)} + E_2^{(p)}\right)\ m^{(p)}\boldsymbol{u}_d^{(p)} + E_1^{(p+1)}\ m^{(p+1)}\boldsymbol{u}_d^{(p+1)}. \end{array} \tag{3.115}$$

In order to solve this system, we derive a θ-scheme, see (Laurent 2002b) and (Laurent 2006), leading, for the mass density, to:

$$\begin{aligned} \frac{m^{(p)\,n+1} - m^{(p)\,n}}{\Delta t} &= -\left(E_1^{(p)} + E_2^{(p)}\right)\left[\theta m^{(p)\,n+1} + (1-\theta)\, m^{(p)\,n}\right] \\ &\quad + E_1^{(p+1)}\left[\theta m^{(p+1)\,n+1} + (1-\theta)\, m^{(p+1)\,n+1}\right] \end{aligned}, \tag{3.116}$$

with $\theta \in [0,1]$. It can be written in the form:

$$A_\theta\, \boldsymbol{m}^{n+1} = B_\theta\, \boldsymbol{m}^n, \tag{3.117}$$

where A_θ and B_θ are bi-diagonal matrices. The diagonal coefficients are $1+\theta\Delta t\left(E_1^{(p)} + E_2^{(p)}\right)$ for A_θ and $1-(1-\theta)\,\Delta t\left(E_1^{(p)} + E_2^{(p)}\right)$ for B_θ, with $p \in \{1,..,N_S\}$, where N_S is the number of multi-fluid sections. The upper diagonal coefficients are $-\theta\Delta t E_1^{(p)}$ for A_θ and $(1-\theta)\,\Delta t E_1^{(p)}$ for B_θ. With $\theta = 1/2$, the scheme offers unconditional stability, preserves the positivity of the mass density in each section and is second order accurate in time. The velocity is not evolving in this purely evaporating step. We finally have the scheme:

$$\begin{aligned} \boldsymbol{m}^{n+1} &= \left(\mathrm{Id} + \frac{\Delta t}{2} C\right)^{-1}\left(\mathrm{Id} - \frac{\Delta t}{2} C\right)\boldsymbol{m}^n, \\ \boldsymbol{m}^{n+1}\ \boldsymbol{u}_d^{n+1t} &= \boldsymbol{m}^{n+1}\ \boldsymbol{u}_d^{nt}, \end{aligned} \tag{3.118}$$

with:

$$C = \begin{pmatrix} E_1^{(1)} + E_2^{(1)} & -E_1^{(2)} & & \\ & \ddots & \ddots & \\ & & \ddots & E_1^{(N_S)} \\ & & & E_1^{(N_S)} + E_2^{(N_S)} \end{pmatrix}. \tag{3.119}$$

The matrices $\left(\mathrm{Id} + \frac{\Delta t}{2} C\right)$ and $\left(\mathrm{Id} - \frac{\Delta t}{2} C\right)$ are constant during the computation, provided Δt is constant, and can then be pre-computed in this case for a higher computational efficiency.

Conclusion

The mathematical study of the multi-fluid system of conservation laws conducted in this chapter allows to design a robust and efficient numerical scheme standing vacuum, high concentration regions and $\delta-$shock singularities. This numerical scheme is used for all the multi-fluid computations conducted in this work, Chpt. 4, Chpt. 5, Chpt. 9, Chpt. 10 and Chpt. 11.

Furthermore, it gives the basis for the analysis of the singularities arising in the monokinetic multi-fluid framework. We recall that these singularities are related to the inability to describe droplet trajectory crossings in Eulerian formulations presuming the velocity

distribution form. A Stokes criterion allowing to predict their creation is provided. This criterion is assessed in Chpt. 4, for Taylor-Green vortices and Homogeneous Isotropic Turbulence configuration. The impact of the singularities at a global modeling level is studied in Chpt. 4.

Chapter 4

Numerical characterization: efficiency, crossing occurence and Lagrangian-evaluated accuracy

Contents

The behavior of the specific numerical method, designed in Chpt. 3, is assessed in this chapter. Two issues are tackled: the behavior of the numerical method in challenging situations, for instance when $\delta-$shocks are created, and the precision of the numerical method, studying its effective order. In order to conduct such precise studies, 2-D vortical flow configurations, as Taylor-Green flow or Homogeneous Isotropic Turbulence, are used. They provide a relatively simple framework, and allow to study the interaction between spray and gas vortices, that is a first step toward more realistic applications. In order to assess singularities treatment, we also use a laminar crossing jets configuration, since the $\delta-$shocks are related to droplet trajectory crossings.
As far as the behavior of the method is concerned, its robustness is assessed in critical situations arising when vacuum as well as high mass density concentration regions are created, and in singularities occurrence. The ability to treat such configurations, is a novelty for Eulerian spray resolution methods. These singularities are further studied. The mathematical analysis conducted in Chpt. 3 is assessed in the chosen configurations. Its ability to predict singularity formation is illustrated. Finally, the global modeling impact of local singularity formation is studied. This study provides the first precise characterization of spray singularities arising when droplet trajectory crossings occur.
Regarding the precision of the method, time-resolved quantitative comparisons with a Lagrangian description are provided. A Direct Simulation Monte-Carlo (DSMC) method is used, since it provides the same level of description as the Eulerian method, Chpt. 2. A converged Lagrangian solution is thus defined, and used to reconstruct Eulerian fields to evaluate the multi-fluid method. These comparisons provide a first validation of the multi-fluid model, needed to assess more complex configurations, Part. IV. Furthermore, they provide original time-resolved and size-conditioned quantitative comparisons between a Lagrangian DSMC and the Eulerian multi-fluid method.
The basis of this study was conducted in the framework of the TIMECOP-AE European project, (de Chaisemartin, Laurent, Massot, and Reveillon 2006), (de Chaisemartin, Laurent, Massot, and Reveillon 2008b). The crossing jet configuration was recently studied in the 2008 Stanford CTR Summer Program, (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009).

4.1 Numerical configurations for multi-fluid characterization

We use 2-D Cartesian gaseous configurations with polydisperse spray introduction to conduct this complete analysis of multi-fluid simulations.

4.1.1 Gaseous flows

We present here the gaseous flows chosen as carrier phase for the spray, in order to characterize the multi-fluid model. As we want to focus on the numerical method evaluation, we choose simple stationary flow field, where we will study the unsteady multi-fluid-modeled spray evolution. Since we are aiming at simulating spray turbulent dispersion, we first choose gas flows allowing to study spray/vortices interaction. This study will be done using Taylor-Green vortices and frozen Homogeneous Isotropic Turbulence. Second, we want to introduce the question of Eulerian spray crossing computation, and to do so we will use steady crossing jets.

Vortical flows

Taylor-Green vortices The first flow field chosen to characterize multi-fluid spray ejection by vortices is the Taylor-Green vortices. These analytical vortices were described in Chpt. 3, Sec. 3.1.1, and the velocity field is plotted in Fig. 3.2. This flow field can lead to high gradients of mass density, due to the spray ejection, and is then well indicated to assess the robustness of the numerical implementation of the multi-fluid model. Furthermore, we will characterize numerically the multi-fluid model limits analyzed in Chpt. 3, Sec. 3.1.2, using the analytical critical Stokes number $\mathrm{St_c}$ introduced in that section.

Homogeneous Isotropic Turbulence In order to carry on with spray/vortices interactions, we choose to use a more complex isotropic homogeneous turbulence. We define this flow as "frozen", as we just take a snapshot from a gaseous homogeneous turbulent computation, and we follow the spray unsteady evolution in this stationary field. This snapshot is taken from a 2-D computation. This turbulent field can clearly not model an industrial device turbulence, but this is not the point of this study. We want here to characterize multi-fluid modeled spray ejection by vortices with a more realistic flow than the Taylor-Green vortices. This could have been done with a 2-D snapshot obtained from a 3-D computation, nevertheless it would not change any conclusion of the study or bring any additional information. We then use in this study a 2-D extracted field, for the sake of practical simplicity. The snapshot is taken from a computation realized by a spectral solver, available in the CORIA ASPHODELE code (Péra 2005; Meftah 2008), in a 128×128 grid. In this computation, a statically stationary turbulence is obtained through a deterministic forcing scheme described in (Guichard, Reveillon, and Hauguel 2004), extended from the ideas of (Overholt and Pope 1998). The idea is to force the computed spectrum to a model spectrum, given in (Overholt and Pope 1998). The integral length scale of the turbulence is defined as:

$$l_0 = \frac{2\pi}{\kappa_0}, \tag{4.1}$$

where κ_0 is the reference wave number, defined from the energy spectrum, corresponding to energy-containing range. In the simulation, κ_0 is given by, (Meftah 2008):

$$\kappa_0 = 7\Delta\kappa, \quad \text{with} \quad \Delta\kappa = \frac{2\pi}{L}, \tag{4.2}$$

where L is the square domain dimension, $L = 1$ mm. The velocity root mean square $u' = 1.5$ ms^{-1} is a reference parameter, as well as the eddy turn-over time $\tau_0 = l_0/u'$. The Reynolds number based on the integral length scale is $\mathrm{Re}_{l_0} = 43$. The vorticity of the frozen field extracted from this computation is plotted in Fig. 4.5 (left).

Crossing jets

Finally, we will study in this chapter the case of crossing jets. This case is extreme from the point of view of Eulerian spray modeling since many Eulerian spray models are based on a mono-kinetic assumption, see Chpt. 2. The "classical" multi-fluid method will fail to simulate the physics of such jets, nevertheless we will assess the numerical behavior of the method in this case. Furthermore this configuration will allow us to introduce a multi-fluid multi-velocity computation, published in (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009), as a preamble to (Kah 2010), devoted to this class of model. The flow is assumed time-independent and uniform in the domain, with a non-zero velocity component in the x-direction. The Reynolds number based on u_0, ν_∞ and x_0 is 1000, where U_0 is the gas velocity and x_0 is the global jet width. We will eventually provide dimensional quantities for illustration purposes. These will be based on a velocity of $u_0 = 1$ ms^{-1}, a length $x_0 = 1.5$ cm, and kinematic viscosity $\nu_\infty = 1.6 \times 10^{-5}$ m^{2}s^{-1}. In addition, we will let $D_0 = x_0/100$, where D_0 is the diameter corresponding to the droplet surface area S_0, and use a typical droplet number of $N_0 = 1000$. The computational domain has a size 6×6, which then corresponds to 9 cm$\times$9 cm in dimensional values. Two spray jets with a width of 0.5 are injected into the domain at a unit dimensionless velocity in two directions forming angles $\pi/4$ and $-\pi/4$, respectively, relative to the gas velocity. The gas flow is uniform, with a x-component U_0 and a zero y-component.

4.1.2 Spray initialization

We describe here the spatial distribution as well as the polydispersion of the spray introduced in the gaseous configurations previously presented.

Spatial repartition

We first take, for vortical flows, a spray uniformly distributed in space, in order to study droplet ejection from the core of the vortices to the edge and to obtain mass concentration at the edges. This configuration is interesting in order to investigate the robustness of the numerical method. We also define for the Taylor-Green vortices configuration a non-uniform space distribution. This repartition is defined thanks to the function $\psi(x) = \sin(x)/x$, for x belonging to the interval $[-\pi, \pi]$. A representation of this repartition can be done representing iso-surfaces of the number density of the droplets as in Fig. 4.1 (left).

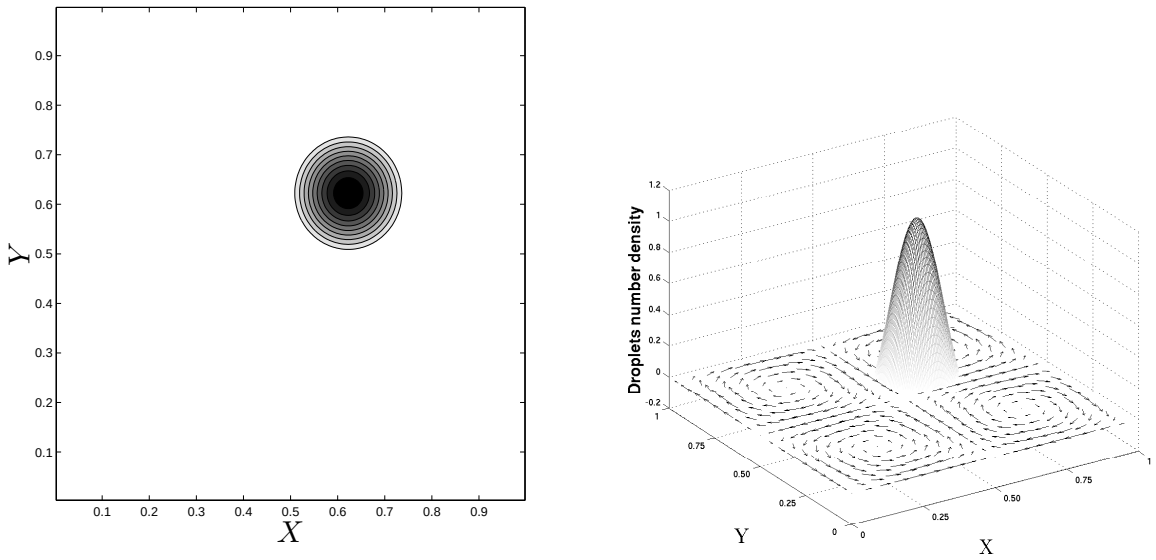


Figure 4.1 – *Droplet initial spatial distribution: non uniform repartition in the domain, (left) number density iso-contours, (right) three-dimensional plot with number density as the third coordinate.*

As a matter of fact, this distribution will allow to analyze numerical diffusion of the method, and to determine the spatial refinements needed. Indeed its gradients are important as we can see in Fig. 4.1 (right) where we realized a three dimensional plot to represent the droplet number density in the domain. The droplets are introduced in the computational domain with a zero initial velocity in both cases.

In the crossing jets simulation, the spray is injected uniformly over the width of the two jets. The spray is injected at the gas velocity.

Size polydispersion

The spray computed are polydispersed. We choose an analytical distribution, with smooth properties and an exponential decrease see (Laurent 2006):

$$f(S) = \frac{(1 + a\ S)\,(1 - S)^2}{b}\ \exp\left(c\left(1 - \frac{1}{(1 - S)^2}\right)\right), \tag{4.3}$$

with:

$$\begin{cases} a &= 8 \\ b &= 1,7 \\ c &= 0.001 \end{cases}. \tag{4.4}$$

This distribution is plotted in Fig. 3.3. We discretize it with 10 to 20 multi-fluid sections. To compute the initial mass density for each section, i.e., the moment of order 3/2 in size of the distribution: $\int_{S_p}^{S_{p+1}} S^{(3/2)} f \mathrm{d}S$, we use a Gauss-Legendre quadrature with ten nodes.

In this section, we want to numerically evaluate the multi-fluid model, we then reduce to simple models. We will use non-evaporating and evaporating spray using a D^2 law. In the crossing jets configuration, we will simulate non-colliding spray.

4.2 Numerical behavior and physical interpretation of multi-fluid singularities

The multi-fluid numerical method has been designed, see Sec. 3.2, to achieve the simulation in extreme situations coming up in multi-fluid method simulation. We thus evaluate numerically the ability of the numerical method to simulate multi-fluid singularities. Furthermore we propose here to assess the mathematical analysis of these singularities proposed in Sec. 3.1.2. We study the ability of the Stokes criterion to foresee singularities formation and we analyze the physical influence of these singularities at a global level.

4.2.1 Multi-fluid singularities computation

Eulerian spray simulation can lead to the creation of vacuum and of high density concentration regions, possibly associated with a shock in velocity. The numerical method used for Eulerian spray simulation must then achieve to simulate such peculiarities.

Vacuum and high density concentration region

The vortical Taylor-Green flow allows us to study the ability to simulate high density concentration due to the ejection of the spray from one vortex, and thus high mass density gradients. We simulate a polydisperse spray, uniformly distributed in space. The polydisperse spray introduced in this flow satisfy the mono-kinetic assumption of the multi-fluid model. To do so, we set a maximal Stokes number of the droplets remaining under the critical value : $\mathrm{St_c} = 1/8\pi$ introduced in Sec. 3.1.2. Otherwise, droplets would leave their initial vortices leading to the creation of crossing zones. We conduct a multi-fluid simulation in a non-evaporating case. We resolve the dynamics of the droplets for the whole range of sizes from zero up to the one corresponding to the critical Stokes number. The results are presented in Fig. 4.2 for a Stokes number close to the critical one, thus exhibiting a rather complete ejection within a time one. Two successive times $t = 0.33$ and $t = 1$ in the non-dimensional setting are presented for droplets whose size correspond to 0.9 time the critical Stokes number with a spatial resolution of 100×100 cells. The numerical scheme does not encounter any difficulties even if the main part of the mass is concentrated in only a few cells.

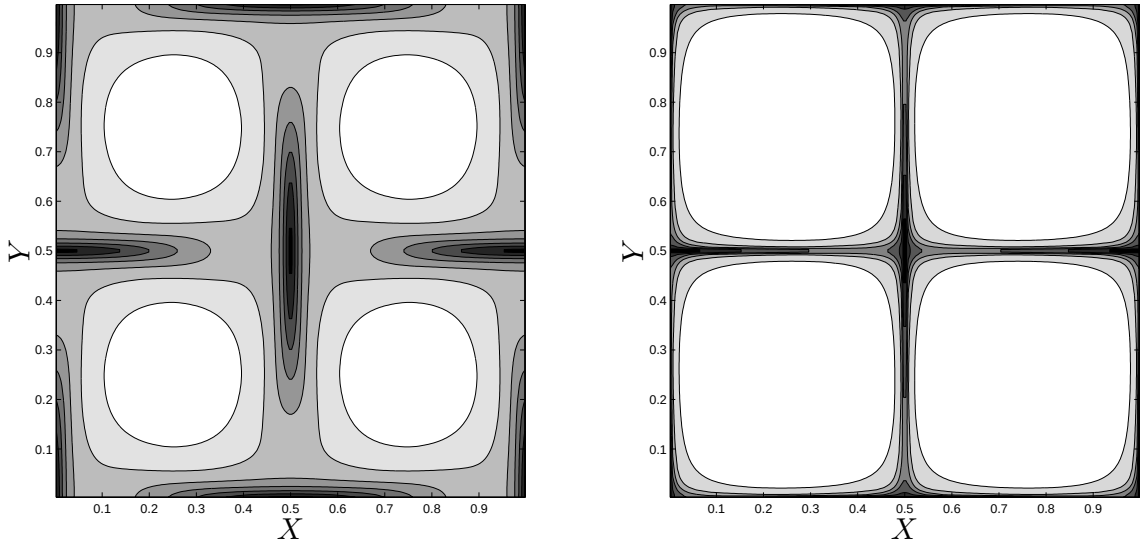


Figure 4.2 – *Snapshots of the droplet mass density spatial distribution at times* $t = 0.33$ *and* $t = 1.0$ *during the ejection from the center of the Taylor-Green vortices to the edges for a section corresponding to* $St = 0.9\,St_c$*. The initial mass density of droplets is uniform with a zero initial velocity distribution.*

The detailed study of the level of refinement needed to precisely reproduce the dynamics of the polydisperse spray will be conducted later on, in Sec. 4.3, throughout detailed comparisons with a Lagrangian solver. This study shows the accuracy of the proposed model and numerical method within the assumptions of the multi-fluid model.

$\delta-$shock **description**

We choose to study $\delta-$shocks computation in the crossing jets configuration. We study two non-colliding, i.e., we take an "infinite Knudsen limit", non-evaporating sprays. If we focus on the equal-size droplets from both spray, or if we take two mono-disperse sprays, a bimodal velocity distribution out of equilibrium arise at the crossing point. This velocity distribution can not be described by the classical mono-kinetic multi-fluid model, able to describe only different-size droplet crossings. A $\delta-$shock is created leading to an artificial averaging of the velocities and to a concentration of the density. We thus switch to an artificial colliding "zero Knudsen limit", see Fig. 4.12, (left). The point we want to make here is not linked to the model impact on the physics of the problem, this point is treated in the next Sec. 4.2.2. Though we underline here the ability of the numerical method to compute, without encountering any difficulty, the $\delta-$shock singularity, as seen on Fig. 4.12, (left). This shows the efficiency of the numerical method to simulate pressureless gas system and validates the scheme derivation proposed in Sec. 3.2.

4.2.2 Trajectory crossings simulation with multi-fluid model

Trajectory crossings simulation in Eulerian spray description leads, in many cases, to modeling problems. Indeed, the description of the, at least, bimodal velocity distribution can not easily be done by that type of model, neither by any classical two-fluid model. The multi-fluid model can take advantage of its polydispersion resolution to describe naturally different-size droplet crossings. Nevertheless, in its original derivation, that we refer as "classical multi-fluid", trajectory crossings of equal-size droplets lead to the outbreak of $\delta-$shocks. This singularity formation can be handled, as shown in Sec. 4.2.1, by the numerical method derived. The impact of the model on the physics is, in the case of two crossing jets with equal-size droplets, so considerable, that the multi-fluid model clearly failed to simulate that type of configuration. Nevertheless, in vortical flows, the outbreak of local crossings, due to the ejection of the spray by the eddies of a vortical flow, needs to be characterized and the model impact evaluated at a global level. Finally, new techniques have been introduced in (Fox 2008) to simulate crossings in Eulerian spray models, and we introduce here their use in a multi-fluid framework.

Vortices ejection: beyond the critical Stokes

In order to characterize the multi-fluid model behavior when crossings occur in a vortical flow, we need to be able to predict such outbreak. We thus study the validity of the critical Stokes criteria introduced in Sec. 3.1.2, for a multi-fluid polydisperse spray computation in a Taylor-Green flow. Then we assess the extension of this analytical criteria, applying the analysis proposed Sec. 3.1.2 to the "frozen isotropic homogeneous turbulence" case. We discuss in that case the impact of local crossing singularities at a global level.

Taylor-Green case If the Stokes number is increased beyond its critical value in the case of a uniform initial droplet mass density spatial distribution, the dynamics of the spray is correctly reproduced until droplets are ejected from one vortex to the other. However, once droplets cross the lattice structure of the vortex edges, a velocity averaging phenomena, already foreseen in a different framework in (Laurent and Massot 2001), leads, by a symmetry argument, to a zero normal mean velocity along the lattice axis. Consequently, a $\delta-$shock arises at this place, that is an important concentration of droplet mass in the surrounding computational cells which are artificially trapped. After some finite time all the mass is concentrated up to numerical diffusion in the cells at the edges of the vortex lattice. Such a behavior can be clearly foreseen due to the symmetrical structure of the initial droplet mass density spatial distribution.

The non-uniform distribution (Fig. 4.1) offers the ability to perform computations with droplets having a Stokes number greater than the critical value of $1/8\pi$ without getting into the same type of behavior (see Sec. 3.1.2). Indeed the exit of the vortex does not lead in this case to frontal crossing of droplets originating from different vortices. This distribution will then allow us to study the behavior of more inertial droplets in a different framework.

We then performed a test case with a Stokes number $\mathrm{St} = 13\,\mathrm{St}_c$, first with a lagrangian computation. The droplets positions are presented in Fig. 4.3, for four times.

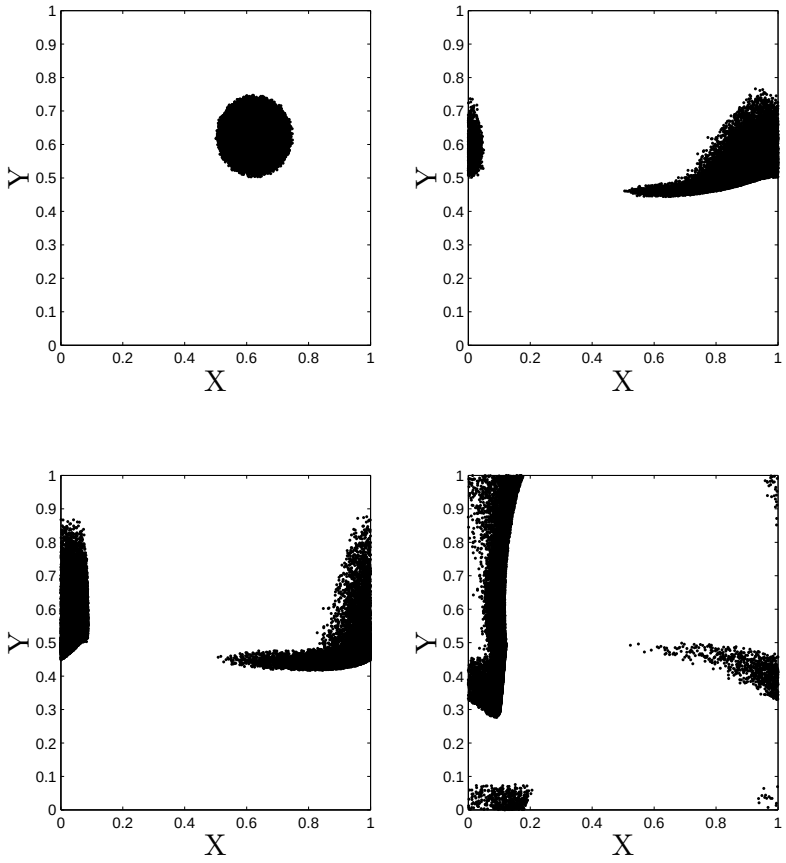


Figure 4.3 – *Lagrangian numerical parcel position at time* $t = 0$, $t = 0.8$, $t = 1$ *and* $t = 1.4$ *in the non dimensional settings for* $St = 13St_c$. *Computation with 50000 numerical parcels.*

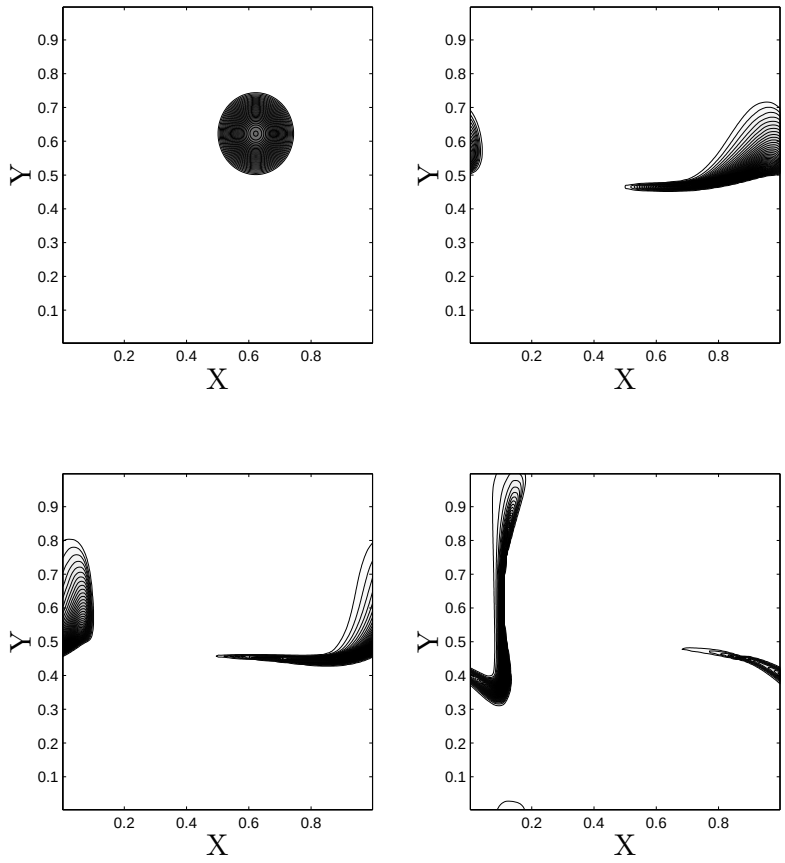


Figure 4.4 – *Eulerian droplet mass distribution evolution at time $t = 0$, $t = 0.8$, $t = 1$ and $t = 1.4$ in the non dimensional setting for $St = 13\,St_c$. 55 iso-contours from 0 up to 1.66-time the maximum of the initial mass density.*

Figure 4.4 shows the dynamics of the ejection of the Eulerian mass density of the spray in the same conditions. In such a case, the spray is ejected from its original vortex, as already predicted by the analytical approach; the numerical simulation reproduces this expected behavior. Nevertheless, there is a tendency of the droplet mass distribution of such size to concentrate in very narrow regions, a phenomenon that is captured by the numerical scheme we have proposed. However, it can be seen that, during the ejection process, the artificial averaging process is already active and leads to such a high segregation of the spray. The behavior predicted by the Williams spray equation and discretized through a Lagrangian solver is slightly different as we can see in Fig. 4.3, for time $t = 1.4$. Thus, in the framework of the Taylor-Green vortices, we have been able to produce a rather clear picture of both the numerical scheme capability and modeling limits of the transport associated with the Eulerian multi-fluid model.

Homogeneous Isotropic Turbulence case Besides, we perform computations beyond the critical Stokes in an Homogeneous Isotropic Turbulence, in order to test the validity of this criteria in more complex gaseous flows than the Taylor-Green vortices. We consider a

frozen homogeneous isotropic turbulence field. We call this "frozen" turbulence since the gaseous flow field is stationary. We represent the iso-vorticity lines of this field in Fig. 4.5, (left). To compute the critical Stokes in this configuration, we use the theory presented in Sec. 3.1.2 and we compute the positive eigenvalue λ_+ to assess the strain rate of the flow (see Fig. 4.5, right), and exhibit regions of high strain rate of the flow. From this field we can compute the distribution of the eigenvalue in the flow field (PDF in Fig. 4.6). The upper value is about 5 and the corresponding critical Stokes is then $\mathrm{St} = 1/20$.

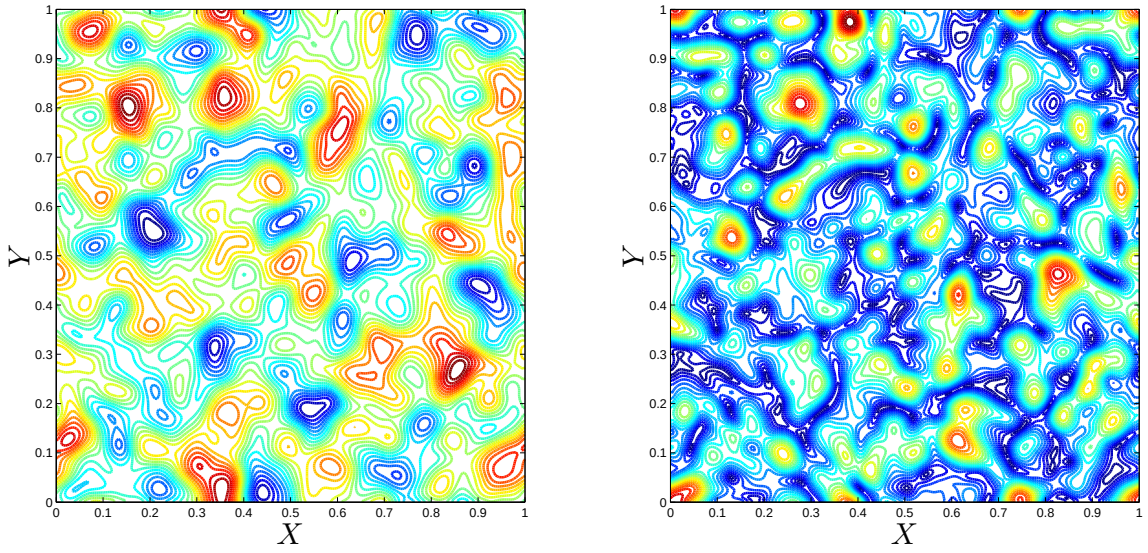


Figure 4.5 – *(left) Frozen gaseous turbulent vorticity field; (rigth) associated strain rate* λ_+*. Grid* 128×128*.*

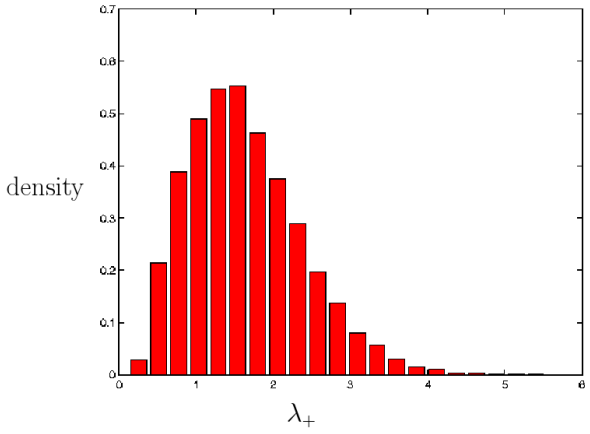


Figure 4.6 – *Distribution of strain rate in the frozen turbulent gaseous velocity field.*

For the sake of legibility, we focus on a small part of the domain, presented in Fig. 4.7 for the turbulent gaseous velocity field and for the normal component of the droplet velocity with respect to the gas one. Important contra-rotating gaseous vortices are present in the

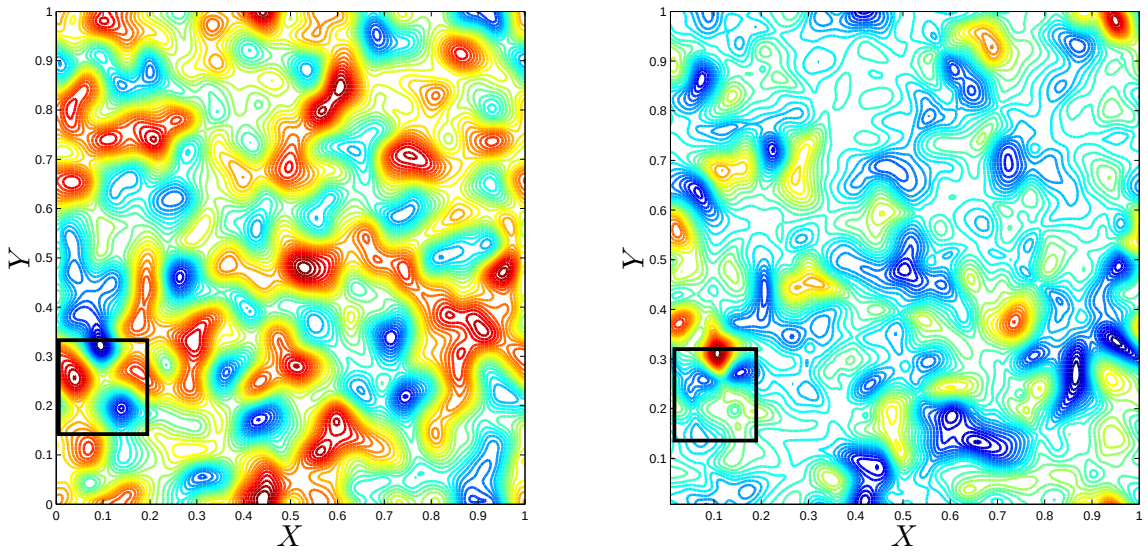


Figure 4.7 – *(left) Frozen turbulent gaseous vorticity field, (right) Eulerian droplet velocity component orthogonal to the gas flow, for the stationary attracting field with 20 iso-contours from -0.01 to 0.015 , at time* $t = 3$ *in the non dimensional setting and for* $St = 0.1$.

considered area which are responsible for the maximal values of the strain rate in the domain (see Fig. 4.8, similarly to the central point for the previous Taylor-Green configuration).

This part of the domain is then relevant to study droplet dynamics evolution with the Stokes number. We thus present a zoom of the gaseous vorticity field in the zone of interest as well as the gaseous velocity field in the same zone in Fig. 4.8, (left), and Fig. 4.8, (right).

Results are then presented for three Stokes numbers corresponding to 0.1 (Fig. 4.9), 0.25 (Fig. 4.10) and 0.5 (Fig. 4.11). The droplet velocity field relaxes, within a non dimensional time equal to a few Stokes number, towards a velocity conditioned on the droplet Stokes number. Since we start with a zero initial velocity field, we can base our study on the invariant droplet velocity manifold at each Stokes number. Consequently, we have plotted the normal component of the droplet invariant velocity field with respect to the gaseous velocity field on the left. On the right of each Figure for the three Stokes numbers, we present the two vector fields, for both the droplets and the gas, with a zoom in the zone of highest strain rate.

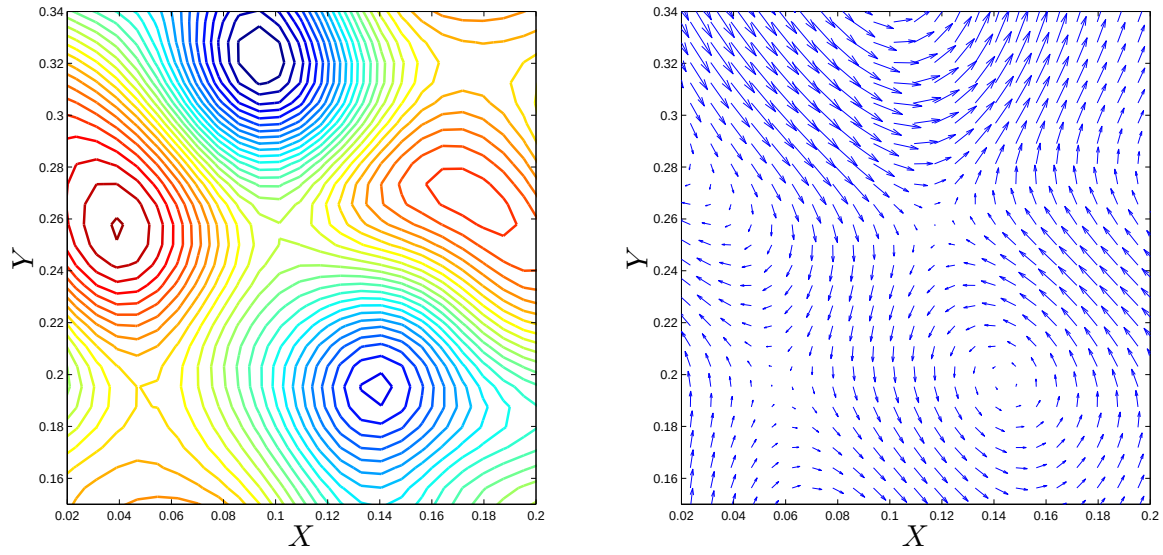


Figure 4.8 – *(left) Frozen turbulent gaseous vorticity field, (right) frozen turbulent gaseous velocity field, at time $t = 3$ in the non dimensional setting. The fields are zoomed in the area of interest.*

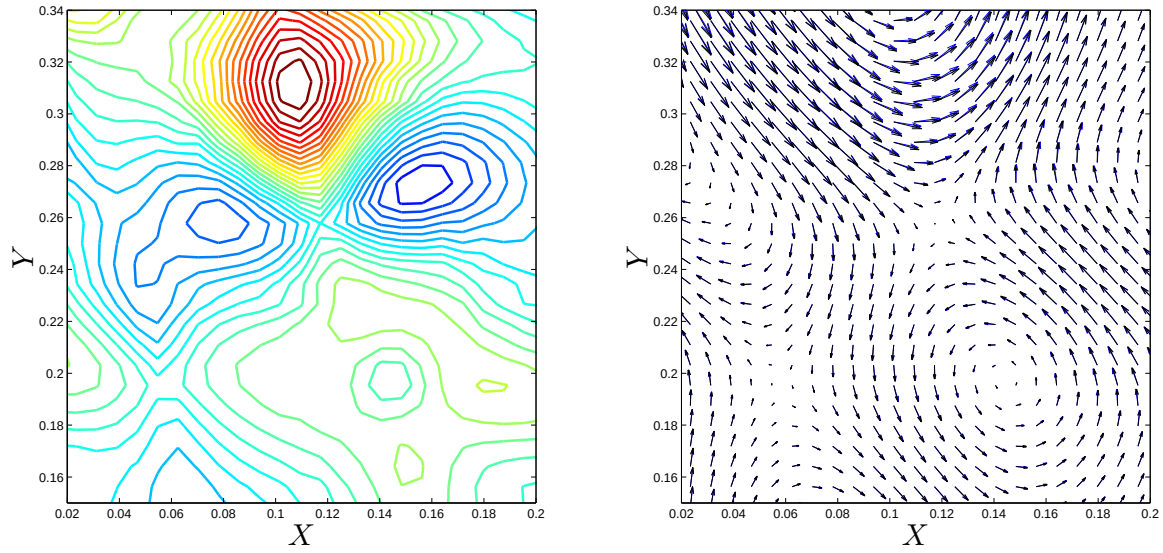


Figure 4.9 – *(left) Eulerian droplet velocity component orthogonal to the gas flow for the stationary attracting field, at time $t = 3$ in the non dimensional setting for $St = 0.1$. 20 iso-contours from -0.011 up to 0.017; (right) vector representation of both droplet (black) and gas (blue) velocities for the same time. Both fields are zoomed in the area of interest.*

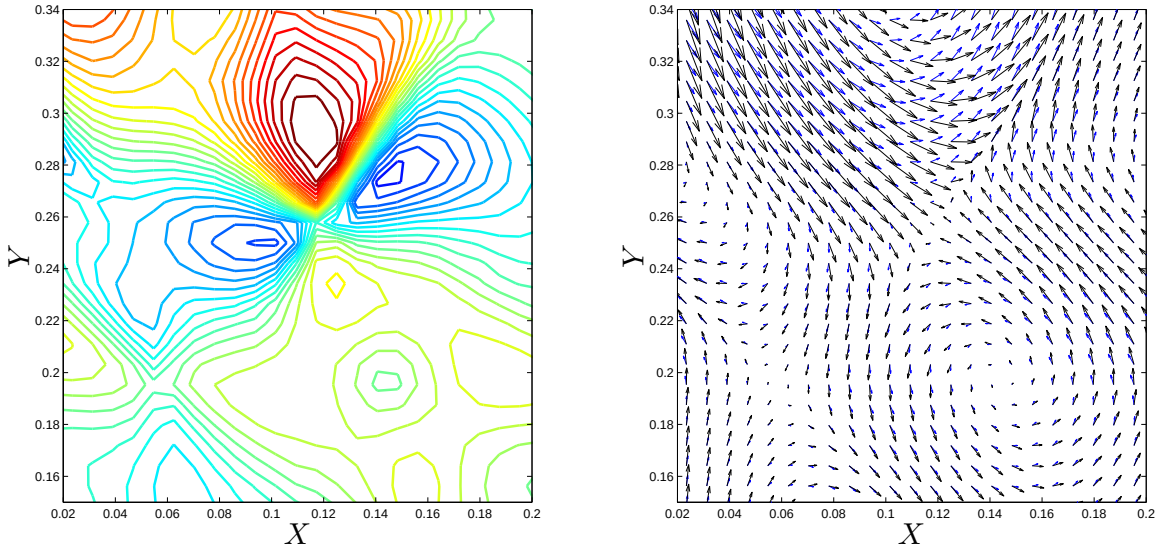


Figure 4.10 – *(left) Eulerian droplet velocity component orthogonal to the gas flow for the stationary attracting field, at time $t = 3$ in the non dimensional setting for $St = 0.25$. 20 iso-contours from -0.08 up to 0.1; (right) vector representation of both droplet (black) and gas (blue) velocities for the same time. Both fields are zoomed in the area of interest.*

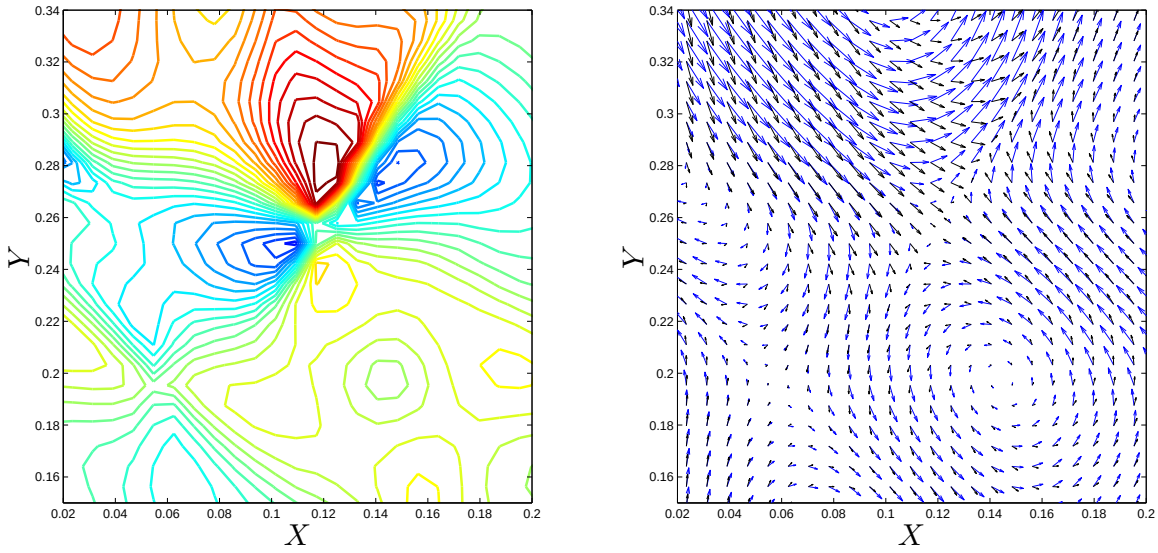


Figure 4.11 – *(left) Eulerian droplet velocity component orthogonal to the gas flow for the stationary attracting field, at time $t = 3$ in the non dimensional setting for $St = 0.5$. 20 iso-contours from -0.14 up to 0.16; (right) vector representation of both droplet (black) and gas (blue) velocities for the same time. Both fields are zoomed in the area of interest.*

When we start with a Stokes number which is 0.1, that is twice the obtained critical Stokes, the amplitude of the normal component of the droplet velocity field with respect with the

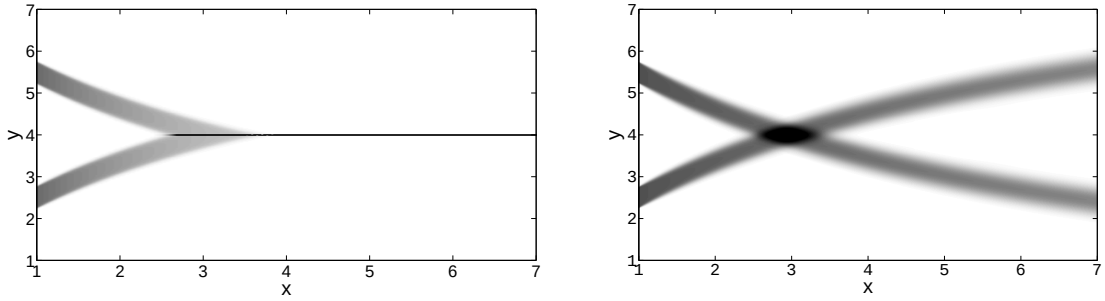
gaseous field can be shown to very be small (see Fig. 4.9 left and right). However, when the Stokes number is close to the critical value, the ejection process from the vortices and the resulting discontinuity is rather weak; such a statement can be shown precisely in the 1D and Taylor-Green vortices cases. Even if there is an impact of this discontinuity on the number density field dynamics, this impact will be very limited.

When the Stokes number reaches 0.25, it is clear from the graphs in Fig. 4.10 that a singularity in the droplet velocity field appears; the range of values of the normal velocity starts growing and, at the scale of the spatial resolution provided for this example in the neighborhood of the highest strain rate point, a discontinuity arises. This velocity discontinuity leads to crossing droplets. Such a behavior will become even stronger for a Stokes number of 0.5, that is ten times the critical value, for which a strong discontinuity has formed in Fig. 4.11, (left), for the normal component of the droplet velocity field. Even in Fig. 4.11, (right), one can observe some arrows having opposite directions in the neighborhood of the highest strain rate point.

We can then conclude that the observed behavior is still the same in this more complex configuration and leads to the same conclusions as before. First the numerical method is able to cope with such difficulties and can capture the singularities appearing in the Eulerian model without leading to unstable solutions. We have to keep in mind that in the case of dilute sprays, the creation of such singularities relates to a failure of the modeling to reproduce the physical behavior of the spray. However, it seems reasonable to think that the limitation of the multi-fluid model associated with the mono-kinetic character of the droplet velocity field will only be of real importance when strong discontinuities appear, that is much above the critical Stokes number. In a such case, we will need extra modeling at the Eulerian level as well as at the kinetic level where eventual interactions between droplets have to be considered depending on the levels of droplet number density.

Crossing jets

We finally assess the extreme situation for crossing outbreak: two crossing jets. We recall here that the sprays are not colliding, i.e., we have a "infinite Knudsen limit". This very challenging configuration for Eulerian model can be simulated by the multi-fluid model as long as different-size droplets are crossing. Nevertheless, as we already shown, an artificial averaging appears, leading to an artificial "zero Knudsen limit", see Fig. 4.12, (right), for equal-size crossings simulation. Recently, the use of the quadrature method of moments in the velocity phase space has provided a closure for non-equilibrium velocity distributions for mono-disperse particles allowing to capture droplet crossing at finite Stokes number with the use of Eulerian model, see (Desjardins, Fox, and Villedieu 2008) and (Fox 2008). As presented in Chpt. 2, these techniques were extended to derive a new multi-fluid multi-velocity Eulerian model, able to simulate efficiently equal-size droplets crossings, see Fig. 4.12, (right).



(a) Multi-fluid. (b) Multi-velocity.

Figure 4.12 – *Mass density for simulation of two crossing jets using the standard multi-fluid approach (left) and the multi-fluid/multi-velocity model (right) for droplets with* $St = 5.29$.

These results are presented in (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009) were colliding jets are also tackled. Furthermore, the multi-fluid multi-velocity model is used in a turbulent free jet with polydisperse evaporating spray injection in (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009).

We thus come up with a new Eulerian model, able to deal with polydispersion as well as non-equilibrium velocity distribution for evaporating sprays. These methods have been recently developed and will be discussed in details in (Kah 2010). In the present work, we mainly focus on the "classical multi-fluid model".

4.3 Multi-fluid and DSMC Lagrangian reference

After demonstrating the numerical efficiency and robustness of the method, characterizing numerically the limits of the model, we evaluate here its accuracy. We decide to assess the precision of the multi-fluid model through detailed quantitative comparisons with Lagrangian simulations, defined as reference solutions. We present first precise and quantitative comparisons between the Lagrangian and the Eulerian descriptions for the Taylor-Green vortices. This analytical gaseous velocity field as well as the polydisperse spray initial condition provide a challenging test case and are representative of the main problems we will encounter in more complex configurations. The first ingredient is to define a procedure to be able to compare the accuracy of both methods; we then use it for both a non-evaporating and an evaporating case. Afterwards, we present the results we obtained concerning comparisons in an Homogeneous Isotropic Turbulence configuration. Finally, we present a first Lagrangian-evaluated accuracy of the multi-fluid multi-velocity method in a crossing jets configuration.

4.3.1 Eulerian Lagrangian comparison procedure

To evaluate differences between the two spray solutions, given a gaseous velocity field, we need to reconstruct Eulerian fields from the statistical information provided by the Lagrangian treatment. We thus need to define an Eulerian grid to perform this reconstruction.

Furthermore, as we want to compare the Lagrangian results to the multi-fluid ones, we want to perform comparisons for various droplet sizes in order to study the ability of the Eulerian method to capture dynamics of droplets of various sizes. In our two dimensional Taylor-Green case, we need a three-dimensional grid to account for space and size discretizations. We have chosen in this case a $100 \times 100 \times 10$ grid, to have a sufficiently detailed description of the fields : 100×100 for the space discretization, and 10 size intervals to be able to study polydispersion. Computing errors on this grid means :

- rebuilding the Eulerian fields for the mass densities in the size intervals from the Lagrangian statistics on this grid;
- averaging, if necessary, the Eulerian multi-fluid results on this very grid.

An example of such a reconstruction is to be found in Fig. 4.13, where a zoom over a quarter of the computational domain, is presented. In order to have statistically converged Lagrangian computation on the Eulerian grid, we choose to compute the Lagrangian with a very high number of numerical particles: 16 Million.

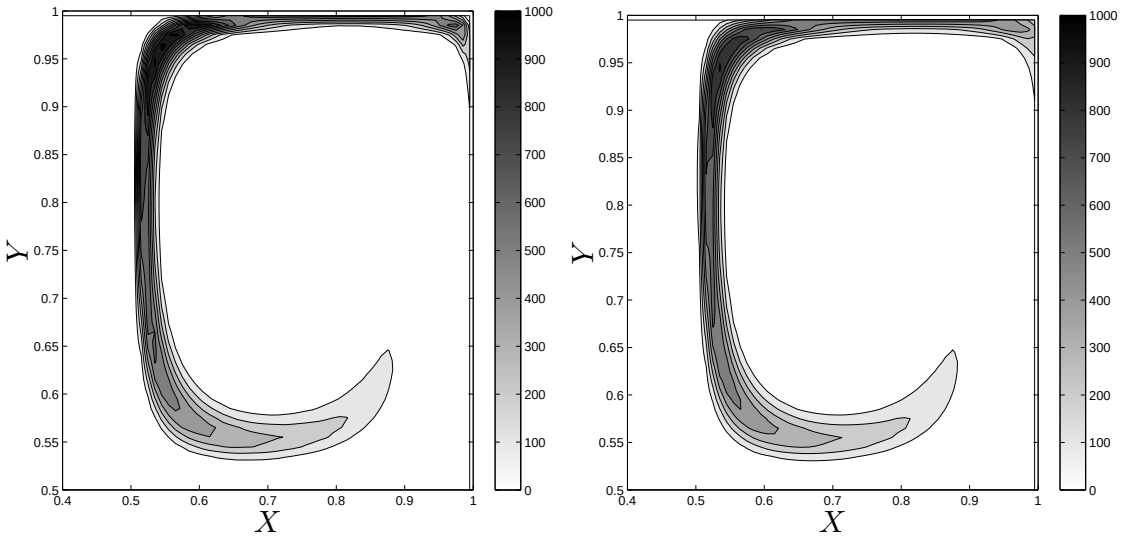


Figure 4.13 – *Zoom on a quarter of the periodic computational domain of the total mass density of droplets at time* $t = 1.5$*, projected on the comparison grid* $100 \times 100 \times 10$ *: (left) Lagrangian reference solution (right) Eulerian with a* $400 \times 400 \times 10$ *grid.*

We evaluate on this grid the global error at the time $t = 1.5$ (i.e. $t = 1.5\,\tau_{gas}$, about one and a half eddy turn over time). The constant by pieces error field for a generic quantity Q, which in the following will be mainly the droplet mass density in one section, is defined for an Eulerian computation by:

$$E_Q^{1.5}(x, y) = Q_{\mathrm{e}}^{1.5} - Q_{\mathrm{l-ref}}^{1.5}, \tag{4.5}$$

where the fields are taken to be constant inside a cell of the comparison grid. The right-hand-side corresponds to the averages over the associated cell of both the Eulerian simulation (subscript e) and the Lagrangian reference simulation with 16 Million particles (subscript $\mathrm{l-ref}$), respectively.

To have an estimate of the relative error over the whole grid, we study the L^1 norm of the relative error on the grid :

$$||E_Q^{1.5}||_1 = \frac{\int_0^1 \int_0^1 |Q_{\mathrm{e}}^{1.5}(x,y) - Q_{\mathrm{l-ref}}^{1.5}(x,y)| \mathrm{d}x\, \mathrm{d}y}{\int_0^1 \int_0^1 |Q_{\mathrm{l-ref}}^{0}(x,y)| \mathrm{d}x\, \mathrm{d}y}. \tag{4.6}$$

The denominator is taken at the initial time, but this has no impact on the non-evaporating case since the global mass as well as the mass in the sections are preserved throughout the calculation. For the evaporating case, this has however an impact and allows to quantify the error relative to the initial amount of mass, which is the one with the main physical sense.

4.3.2 Eulerian and Lagrangian sprays in Taylor Green vortices

Non evaporating spray

To illustrate comparisons between the Eulerian and the Lagrangian methods previously described, we present a study of the error between the Lagrangian reference case with 16 Million particles and the Eulerian cases, to show that both methods converge towards the same results. The chosen time for the comparison is $t = 1.5$, for which the total mass density is presented in Fig. 4.13. It exhibits a high concentration of the inertial droplets around the vortex and creates very high gradients. It is thus a really challenging test case as far as Eulerian models are concerned, irrespective of the evaporation process. We now discuss the error plots presented in Fig. 4.16 to Fig. 4.15 where the details of the comparisons are conducted for three representative sections illustrating the influence of the droplets inertia on their behavior. The pictures represent the evolution of the logarithm of the relative error between Eulerian multi-fluid cases and the reference Lagrangian computation versus the logarithm of the space discretization step of the Eulerian computation. For convenience, we have presented the actual values of the error and of the space discretization step on the two axis, but the points on these axis correspond to the logarithms of these values. As we mentioned in Sec. 3.2, the numerical method used for the multi-fluid model is second order in space, explaining the line with a slope equal to two in all the pictures. The plots show three different levels of space refinements for the Eulerian computation: 100×100, 200×200 and 400×400. In order to see the influence of the refinement in size of the multi-fluid model, we plot three curves in each Figure, showing three size discretizations: 10 sections, 20 sections and 30 sections.

Concerning the droplets of intermediate size presented in Fig. 4.14, the Eulerian computation converges towards the Lagrangian reference case with almost second order. Furthermore, ten sections are enough for the size discretization since the size refinement does not have any impact on the global error. The effect of the size discretization step refinement in this non-evaporating case is purely a finer description of the velocity distribution as a function of droplet size for a given location and time. Consequently, in the range of Stokes number associated to these droplets of intermediate size, the size-dependence of the velocity field is not strong enough to require a finer discretization and the dynamics of the droplets are correctly reproduced by the 10 sections case.

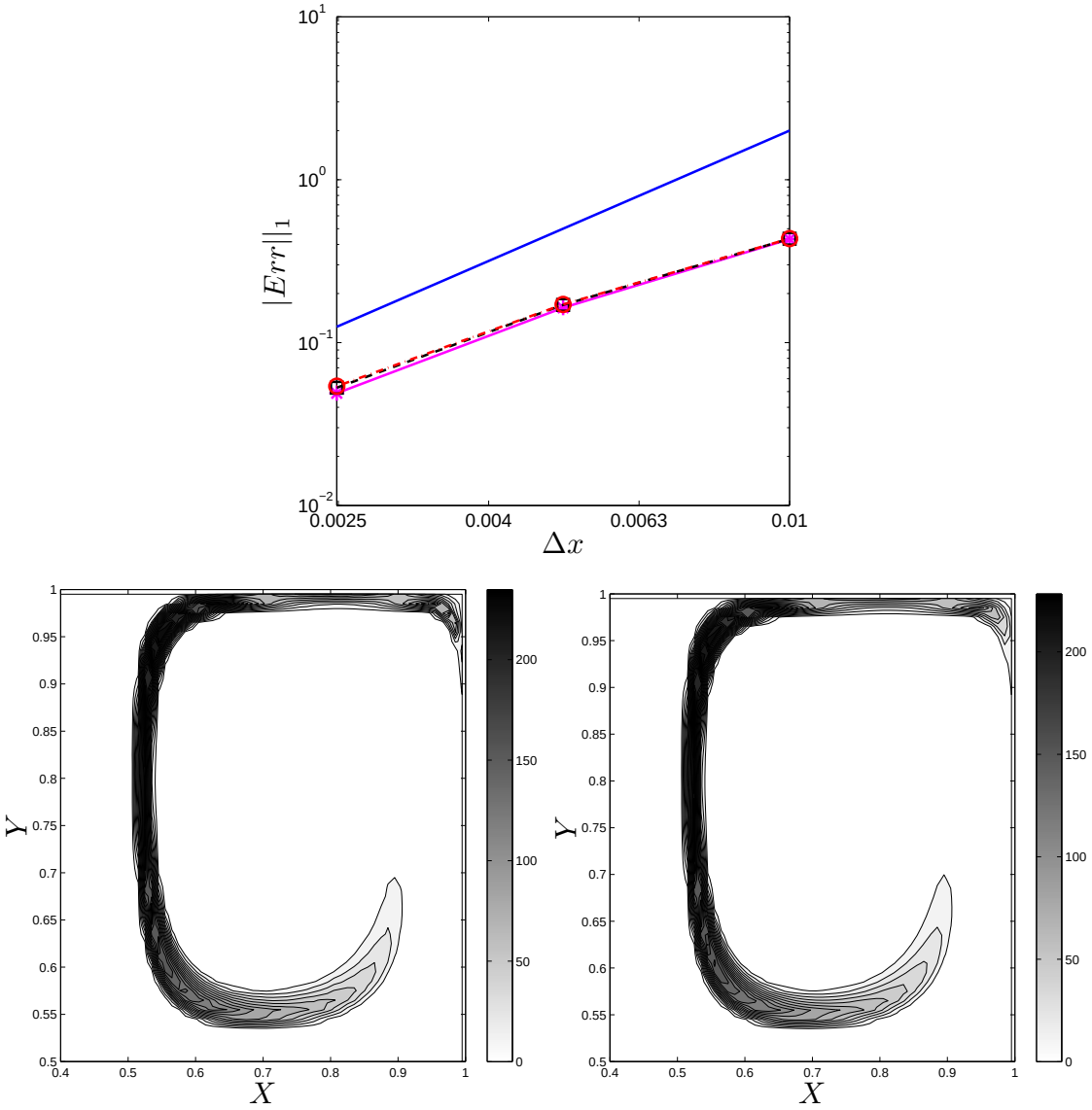


Figure 4.14 – *(top) Logarithm of L^1 norm of the error between Eulerian simulation for medium size droplets (fifth section $St = 0.41\,St_c$) for various Δx and the Lagrangian reference solution at time $t = 1.5$, projected on the comparison grid $100 \times 100 \times 10$: slope two line (blue solid line), 10 sections (solid line), 20 sections (dashed line) and 30 sections (dotted-dashed line). (bottom) Corresponding droplet mass density with 10 iso-contours from 0 up to 1.44-time the maximum of the initial mass density in the fifth section : (left) Lagrangian reference solution, (right) Eulerian computation with a $400 \times 400 \times 10$ grid.*

For more inertial droplets, Fig. 4.15, the order of convergence is weaker and it is not improved by the refinement in section. As a matter of fact, this is due to the high gradient appearing for this class of droplets; indeed the inertial droplets are quickly ejected from the vortex and form high concentration regions and therefore very high gradients (most of the mass is concentrated in a few cells). In this case, the numerical method reduces its order of precision to first order to deal with such concentrations and associated gradients, thanks

to the use of a slope limiter we presented in Sec. 3.2.2. When we increase the number of points we decrease these gradients and we therefore increase the order of the method, as represented in the Fig. 4.15.

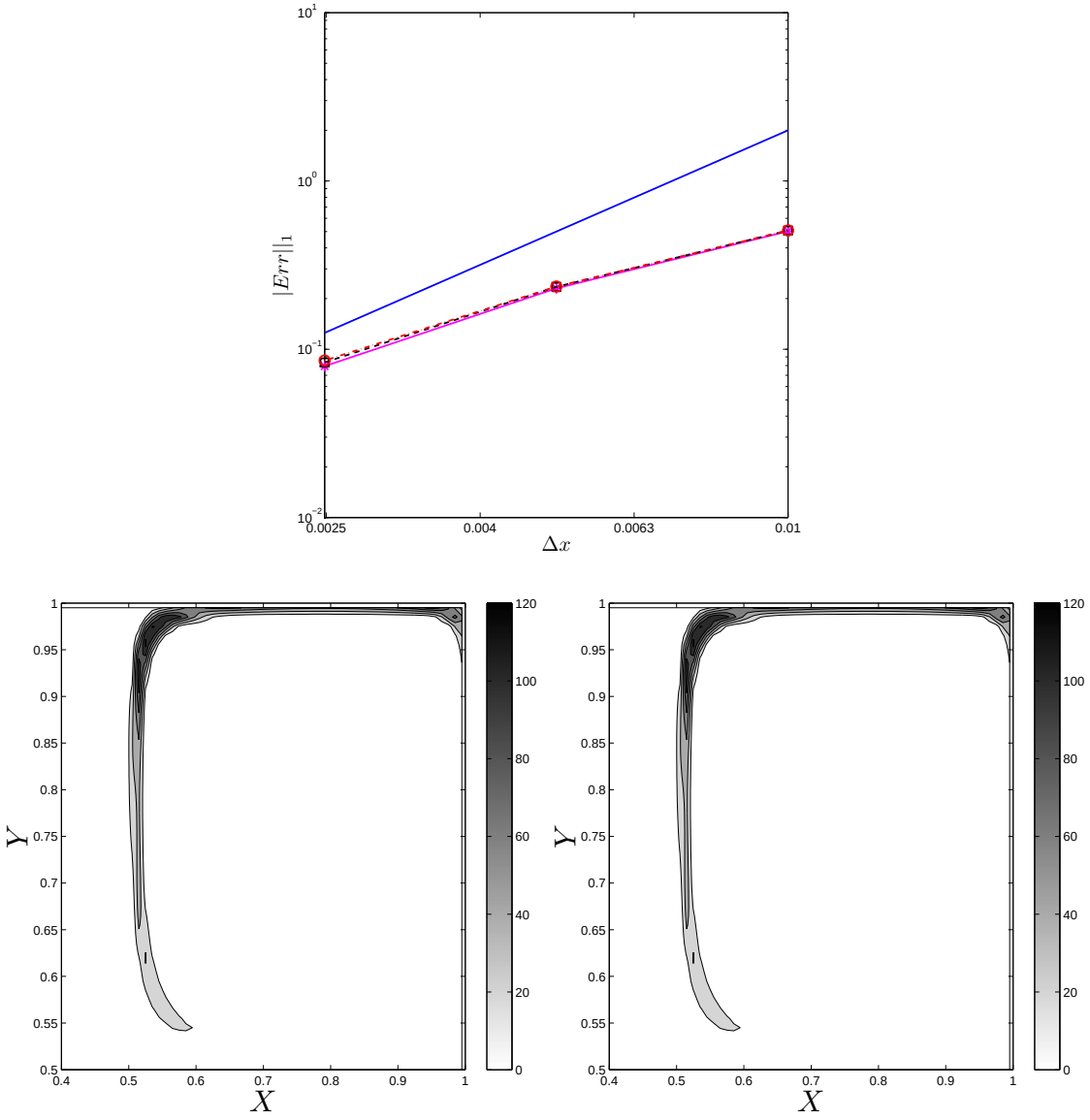


Figure 4.15 – *(top) Logarithm of L^1 norm of the error between Eulerian simulation for inertial droplets (ninth section $St = 0.78\,St_c$) for various Δx and the Lagrangian reference solution at time $t = 1.5$, projected on the comparison grid $100 \times 100 \times 10$: slope two line (blue solid line), 10 sections (solid line), 20 sections (dashed line) and 30 sections (dotted-dashed line). (bottom) Corresponding droplet mass density with 10 iso-contours from 0 up to 2.43-time the maximum of the initial mass density in the ninth section : (left) Lagrangian reference solution, (right) Eulerian computation with a $400 \times 400 \times 10$ grid.*

As far as the small droplets are concerned, Fig. 4.16, the behavior is different from the previous ones. Indeed we notice an important influence of the droplet size refinement in

the rate of convergence. In this case with droplets of low inertia, the build up of the overall droplet mass density spatial gradients does not build up to strongly so that the change in order is not due to excessive gradients created by the droplet rapid ejection from the vortices; nevertheless the droplet velocity is close to the gas one, but the velocity difference between spray and gas can be shown to be linear, in first approximation, with respect to the Stokes number and, thus one can guess that the velocity conditioned by droplet size is a linear function of droplet surface around zero. The assumption of the multi-fluid (see Sec. 2.2.3) imposes a velocity which is constant as a function of size inside a section and thus it is not able to capture this effect. Then we have to switch to 20 sections to properly predict the dynamics of droplets with very low inertia. Once refined in size, the Eulerian computation converges towards the Lagrangian reference with an order of about 2.

Both descriptions are thus presenting a very good agreement as we can see quantitatively by analyzing the error levels. The L^1 norm of the error is around a few percent for a Eulerian simulation with a $400 \times 400 \times 10$ discretization as summarized in Table 4.1. Such an error can be thought of as rather high. It can be explained in the following way. We have chosen to present the relative error in reference of the initial total mass introduced in the numerical simulation, we have a small amount of mass concentrated in a narrow region of the domain which gets even more concentrated due to the ejection process by the vortex and finally, we have extreme gradients in order to test the influence of the numerical diffusion of the second order in space numerical method. If, however we decided to plot the absolute error only considering that the mass density of droplets is reaching one at its maximum, we would end up with an error of about one per a thousand. Such a statement can be observed more qualitatively, plotting the iso-contours of the mass density for the three sections studied in Fig. 4.16, Fig. 4.14 and Fig. 4.15. Dynamics of the droplet of various sizes are very well predicted, even if the chosen test case is extremely challenging and leads to the presented level of errors.

Droplet size	Global error norm $\|\|E_m^{1.5}\|\|_1$
$\mathrm{St} = 0.14\,\mathrm{St}_c$	8%
$\mathrm{St} = 0.41\,\mathrm{St}_c$	5%
$\mathrm{St} = 0.78\,\mathrm{St}_c$	8%

Table 4.1 – *Computation of* $||E_m^{1.5}||_1$*, norm of the global error at time* $t = 1.5$*, for the mass density of three sections of droplets in size: small droplets (*$St = 0.14\,St_c$*), medium size droplets (*$St = 0.41\,St_c$*) and inertial droplets (*$St = 0.78\,St_c$*).*

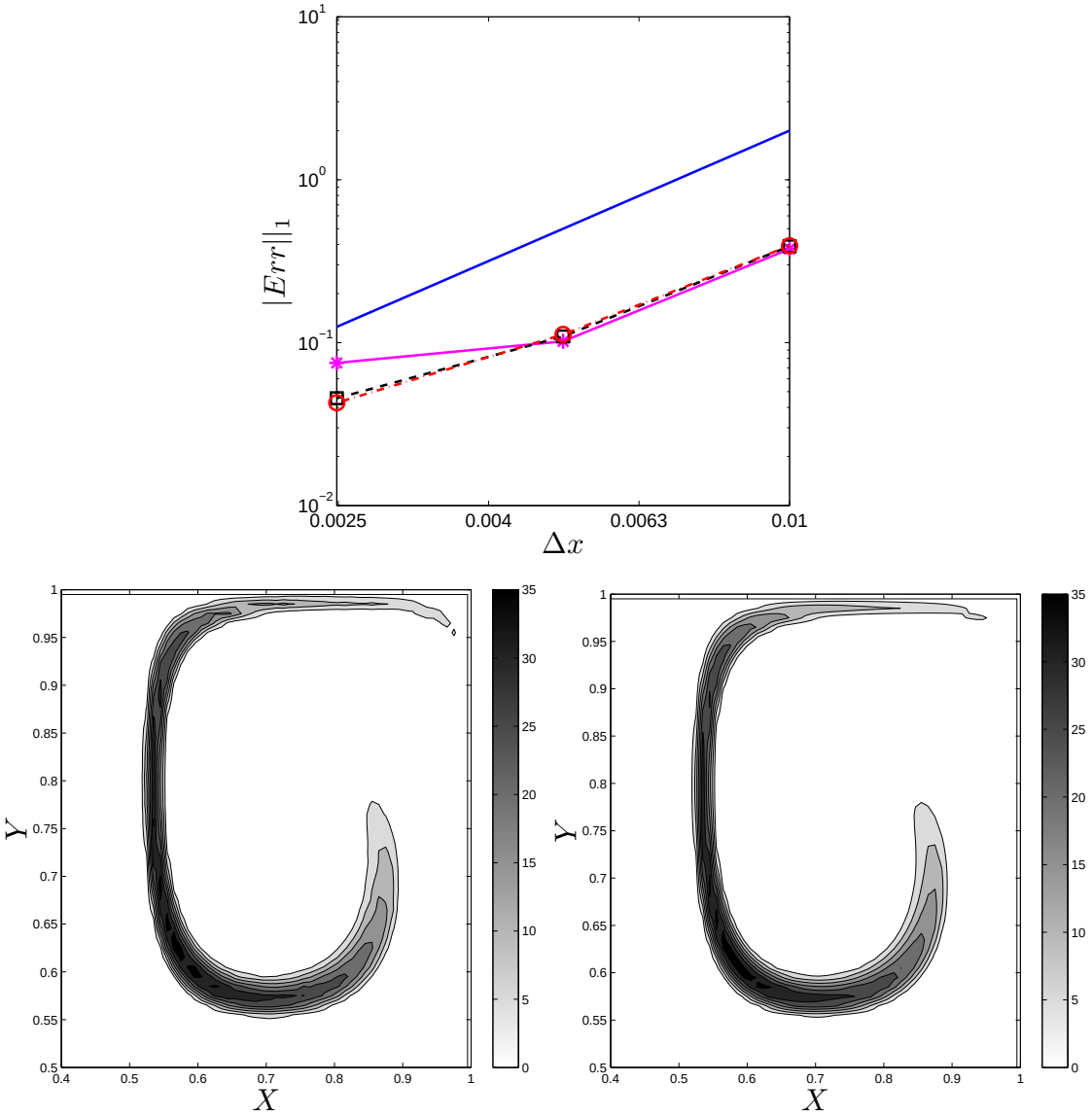


Figure 4.16 – *(top) Logarithm of* L^1 *norm of the error between Eulerian simulation for small droplets (second section* $St = 0.14\,St_c$*) for various* Δx *and the Lagrangian reference solution at time* $t = 1.5$*, projected on the comparison grid* $100 \times 100 \times 10$ *: slope two line (blue solid line), 10 sections (solid line), 20 sections (dashed line) and 30 sections (dotted-dashed line). (bottom) Corresponding droplet mass density with 10 iso-contours from* 0 *up to* 1.44*-time the maximum of the initial mass density in the second section : (left) Lagrangian reference solution, (right) Eulerian computation with a* $400 \times 400 \times 10$ *grid.*

Evaporating case

Our interest being in combustion applications, we perform similar comparisons in the evaporating case, the evaporation being described by a D^2 law in both Eulerian and Lagrangian descriptions (see Sec. 2.2.3). We still refer to a Lagrangian computation with 16 Million

particles. To do so, we have chosen a low evaporation speed to preserve a relatively high number of particles in our reference computation. We then chose an evaporation constant $E_v = 1/15$ (introduced in Sec. 2.2.3), thus if we look at non dimensional $t = 1.5$, 10% of the mass will be evaporated. We present in this case the errors already defined for the non evaporating case between Lagrangian and Eulerian descriptions. We study the same mesh refinements for the Eulerian method, in space as well as in size, and we study the behavior of various droplet sizes.

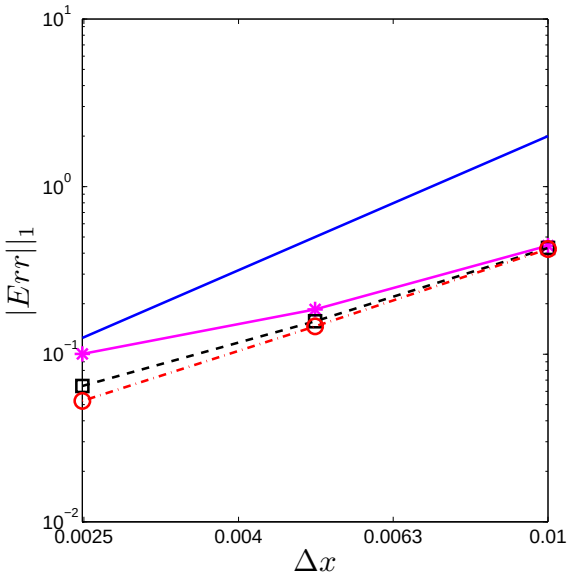


Figure 4.17 – *Logarithm of L^1 norm of the error between Eulerian simulation for small droplets (second section $St = 0.14\,St_c$) for various Δx and the Lagrangian reference solution at time $t = 1.5$, projected on the comparison grid $100 \times 100 \times 10$ in the evaporating case : slope two line (blue solid line), 10 sections (solid line), 20 sections (dashed line) and 30 sections (dotted-dashed line).*

We see in Fig. 4.17 and Fig. 4.18 that the behavior of small and intermediate droplets is the same as in the non evaporating case: the Eulerian method converges towards the Lagrangian reference solution with the expected second order. On the other hand, for the inertial droplets, the refinement in space nearly do not change the error value, only modified by the refinement in sizes, as we can see in Table 4.2.

As we introduced in Sec. 2.2.3, the multi-fluid model for evaporation is only first order accurate in the size discretization step (Laurent 2006). This explain why we do not see the effect of the space refinement, hidden by the need to increase the number of cells/sections in the size phase space. This first order of the evaporation model can be illustrated by the same type of log/log Figure as we presented so far, but taking the logarithm of the size refinement, at fixed space discretization varying between $\Delta x = 1/100$ and $\Delta x = 1/400$. This is done in Fig. 4.19.

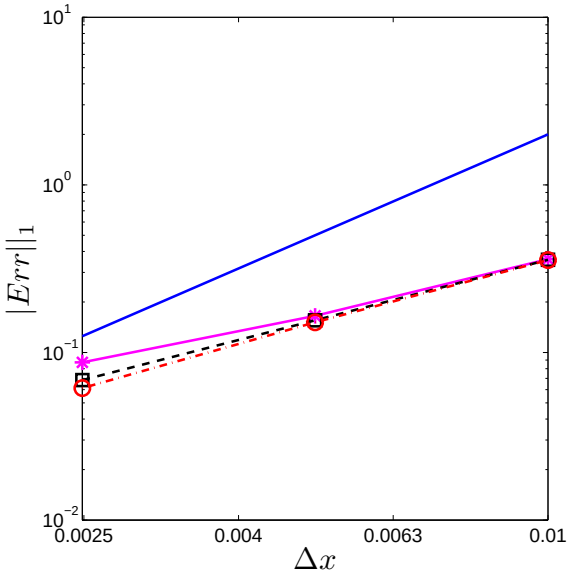


Figure 4.18 – *Logarithm of L^1 norm of the error between Eulerian simulation for medium size droplets (fifth section $St = 0.41\ St_c$) for various Δx and the Lagrangian reference solution at time $t = 1.5$, projected on the comparison grid $100 \times 100 \times 10$ in the evaporating case : slope two line (blue solid line), 10 sections (solid line), 20 sections (dashed line) and 30 sections (dotted-dashed line).*

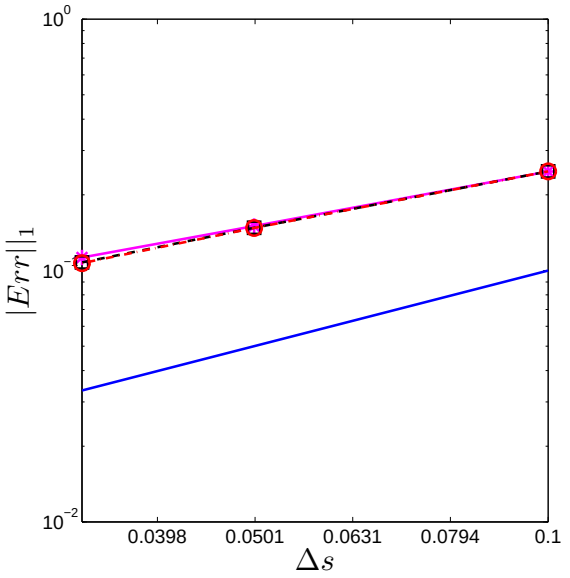


Figure 4.19 – *Logarithm of L^1 norm of the error between Eulerian simulation for inertial droplets (ninth section $St = 0.78\ St_c$) for various ΔS at fixed Δx and the Lagrangian reference solution at time $t = 1.5$ projected on the comparison grid $100 \times 100 \times 10$ in the evaporating case : slope one line (blue solid line), $\Delta x = 1/100$ (solid line), $\Delta x = 1/200$ (dashed line) and $\Delta x = 1/400$ (dotted-dashed line).*

$1/\Delta x$	Error 10 Sections	Error 20 Sections	Error 30 Sections
100	25%	14%	10.5%
200	25%	15%	11%
400	25%	15%	11%

Table 4.2 – *Evolution of L^1 norm of the relative error between the reference Lagrangian solution and Eulerian multi-fluid for various refinements in space and size, projected on the comparison grid* $100 \times 100 \times 10$*, and for inertial droplets* ($St = 0.78\, St_c$)*. The results are presented at time* $t = 1.5$*.*

This drawback of the multi-fluid method was already noticed in (Laurent, Massot, and Villedieu 2004) and can be partially avoided by using a second order method to describe the evaporation as described in (Laurent 2006) and (Dufour and Villedieu 2005). These techniques are not assessed in this work because of the recent developments of multi-fluid methods based on high order moment method in size, (Massot, Laurent, Kah, and de Chaisemartin 2009; Kah 2010), yielding a very efficient description of the evaporation process. As a conclusion, we have shown in both the evaporating and non-evaporating cases the ability of the multi-fluid model to accurately approximate the Lagrangian reference solution, and thus the dynamics of droplets of various sizes coupled to the evaporation process. Even if some improvement of the description of the evaporation process is to come, we can still define an equivalent level of accuracy for both descriptions and thus come up with relevant information for computational cost analysis, Chpt. 9.

4.3.3 Eulerian and Lagrangian sprays in Homogeneous Isotropic Turbulence flows

The Taylor-Green configuration can be considered to be a challenging test in the sense that it involves very high gradients of the number density function. Thus the results obtained in the previous part of this section can be considered as reliable in terms of accuracy of the methods. However, to show the ability of the Eulerian model to capture turbulent configurations, we also provide some more qualitative comparisons in the case of a frozen Homogeneous Isotropic Turbulence. This second configuration will be proved to be useful in terms of computational cost comparisons. The vorticity of the selected gas field is represented in Fig. 4.20, (left).

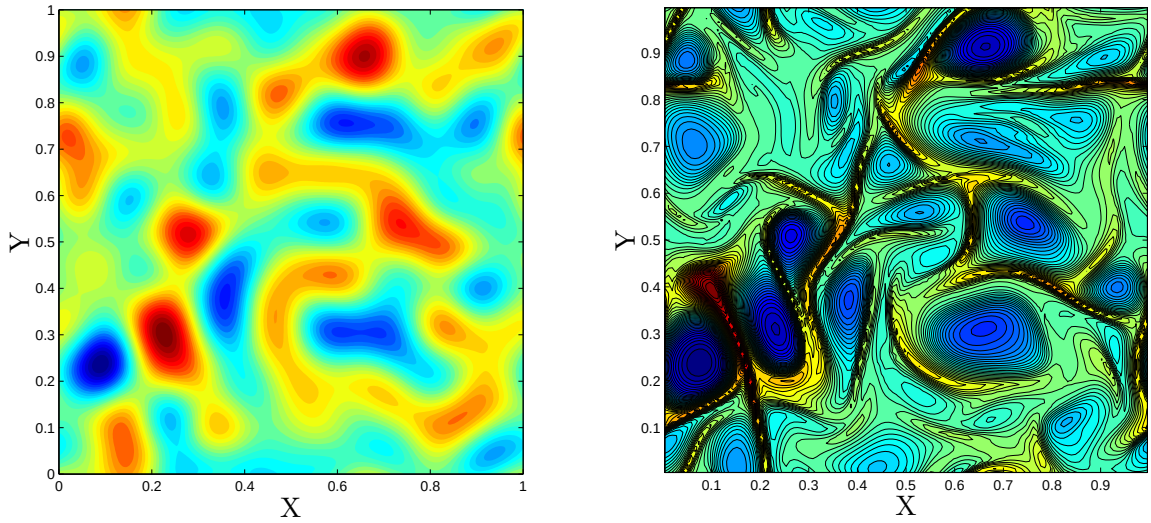


Figure 4.20 – *(left) Frozen turbulent gaseous vorticity field; (right) Eulerian total mass density of the spray at time* $t = 1.5$ *with 50 iso-contours between 0 and 25-times the initial uniform level.*

We performed Lagrangian and Eulerian computations with a polydisperse spray and an initial uniformly distributed NDF as far as space is concerned. We present in Fig. 4.21 the Eulerian and Lagrangian total mass distribution at the non dimensional time $t = 1.5$. Computations are done with 4 Million particles for the Lagrangian and a $256 \times 256 \times 10$ grid for the Eulerian. We represent, in this figure, the total mass distribution, i.e sum over all the size intervals.

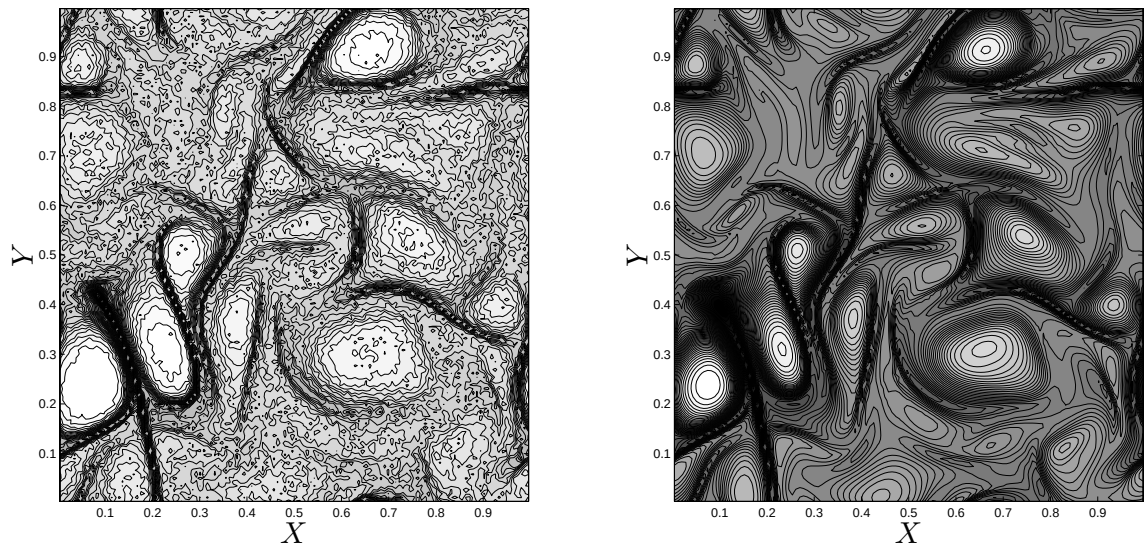
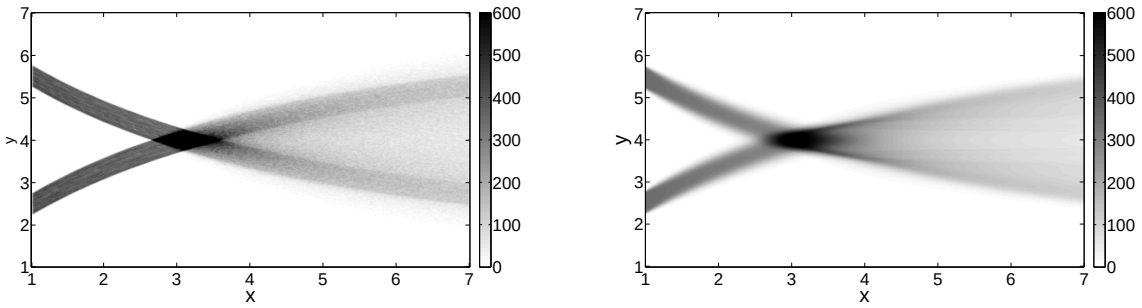


Figure 4.21 – *Total mass density spatial distribution at* $t = 1.5$*, projected on a* $128 \times 128 \times 10$ *comparison grid with 50 iso-contours between 0 and 25-times the initial uniform level which is the same for both computations: (left) Eulerian with a* $256 \times 256 \times 10$ *grid; (right) Lagrangian with 4 Million particles.*

These first results show a very good qualitative agreement between the Lagrangian and Eulerian descriptions. We can observe the segregation effect of the vortices on the spray mass density distribution, the structure of which corresponds clearly to the iso-vorticity lines of the gaseous phase. The ejection of the droplet mass density from the vortical structures generates region of high density and creates very large density gradients, which are both described precisely by the two methods. This is not the purpose to reproduce the same kind of analysis as before for this second case. It is shown for illustration purposes in terms of precision since it is not as challenging as the Taylor-Green configuration.

4.3.4 Eulerian and Lagrangian sprays in crossing jets

In order to have a first evaluation of the multi-fluid multi-velocity model, we present here comparisons with Lagrangian computations in a crossing jet configuration. The gas velocity and the spray initialization have already been described in Sec. 4.1. However, the sprays computed here are mono-disperse and weakly collisional. In this manner some droplets of the jets follow their original trajectories as if they did not see the other ones, whereas the other droplets undergo collisions and are deviated from their original trajectories. The collisions are described as hard sphere Boltzmann collisions. The Bolztmann collision multi-fluid modeling presented in Chpt. 2 is here extended to the multi-fluid multi-velocity model, (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009).



(a) Lagrangian. (b) Eulerian.

Figure 4.22 – *Mass density for the simulations of two crossing mono-disperse jets using the DPS approach averaged and projected onto the Eulerian grid (a) and the multi-fluid/multi-velocity model (b) for droplets with* $St = 5.29$ *undergoing collisions.*

The statistical Lagrangian results are not obtained here through a computation with a high number of numerical parcels, but with an average over a high number of time steps, the configuration being statistically stationary after a transitory regime. The mass density comparisons, Fig. 4.22, shows a good agreement between the two descriptions of the crossing-collisional sprays. The limit between non colliding and colliding droplets does not appear in the Eulerian simulation as clearly as in the Lagrangian one. This is due to the numerical diffusion induced by the transport scheme used for the multi-fluid multi-velocity model, see Chpt. 2. However, the level of mass density are very similar, showing a very good description of the collisions.

These computations are presented in (Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009), along with bidisperse crossing sprays comparisons.

Chapter 5

Eulerian multi-fluid modeling and simulation versus experimental measurements

Contents

The detailed numerical multi-fluid characterization performed in Chpt. 4, studying multi-fluid's assumptions and model impact in vortical flows, enables to design configurations where the multi-fluid model is expected to give accurate results. We then decide to design an experimental configuration allowing multi-fluid validation through experience/simulation comparisons. For such a validation to be achieved, we need to be able to capture size-conditioned dynamics in the experiment.

We consider acoustically pulsated free jets with a polydisperse spray injection in an axisymmetric configuration. The acoustic excitation creates periodical large vortical structures, which are representative of the dynamics of gaseous flows in more complex configurations, and a strong interaction with the injected spray. In this context, we provide a series of detailed experimental measurements using laser diagnostics; we analyze droplet size distributions, associated size-conditioned dynamics and evaporation, for three liquids, leading to various droplet preferential concentrations due to the vortices. Besides the achievements in terms of diagnostics, we use the Eulerian multi-fluid model and dedicated robust numerical schemes, provided in Sec. 3.2, in order to conduct simulations in the chosen configuration. Detailed comparisons between numerical simulations and experimental measurements are provided in terms of spray velocity and number density. Size-conditioned preferential concentration of both non-evaporating and evaporating sprays are observed and reproduced by the numerical simulations, which eventually yields a validation of the proposed model.

The ability to analyze size-conditioned dynamics represents an important novelty, as much from the experimental and the numerical, since a Eulerian method is used, point of view.

This work has been done in collaboration with L. Fréret with regard to the simulation part. The experiment has been realized by C. Lacour, D. Durox and S. Ducruix, in EM2C laboratory. Global comparisons were first presented in the 2007 International Conference on Multiphase flows, (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007). The size-conditioned comparisons described in this chapter have been published in the 32$^{\text{nd}}$ International Symposium on Combustion, (Freret, Lacour, de Chaisemartin, Laurent, Massot, Ducruix, and Durox 2008).

5.1 Pulsated free jet with spray injection

Crow and Champagne demonstrated that a circular jet has a natural instability at a frequency corresponding to the preferred mode of the jet (Crow and Champagne 1971). This instability can be characterized by the Strouhal number Sh, defined by

$$\mathrm{Sh} = \frac{f D}{U_b}, \tag{5.1}$$

	D_{10}	D_{32}	$E_v(D_{32})$	$\mathrm{St}(D_{32})$
decane	26	36	93	0.12
methanol	20	32	3.4	0.13
heptane	20	40	6.8	0.15
	$E_v(18)$	$\mathrm{St}(18)$	$E_v(60)$	$\mathrm{St}(60)$
methanol	0.96	0.037	12	0.43

Table 5.1 – *Mean size distribution and Sauter mean diameter (in μm) of decane, methanol, heptane and their corresponding evaporation numbers. In case of methanol, values of evaporation numbers for diameters of 18μm and 60μm.*

based on the excitation frequency f, on the jet diameter D and on the mean axial bulk velocity U_b. According to (Crow and Champagne 1971; Birbaud, Durox, Ducruix, and Candel 2007), the jet strongly responds to excitations around the preferred mode $\mathrm{Sh} \simeq 0.4$, but is still very sensitive up to $\mathrm{Sh} = 1.5$. Consequently, such a jet is an ideal candidate for a specific study on the behavior of a spray in a flow exhibiting zones of high vorticity. Thus the retained configuration is similar to the one in (Pichard, Durox, and Ducruix 2005), in a non-reactive excited jet, laden by droplets. The range of selected Reynolds numbers is of the order of 1000 and yields a laminar gas flow at the nozzle outlet.

In order to allow easy comparisons between numerical simulations and experimental results, we work with dimensional quantities; however, the physics of the problem are governed by two non-dimensional numbers. The reference time $\tau_f = 1/f$ is the acoustic time corresponding to the frequency f imposed to pulsate the jet. The reference velocity U_0 is the mean axial velocity at the nozzle exit. A reference length can be deduced from these two quantities, which is characteristic of the coherent structure displacement. Characteristic droplet surface S_0 corresponds to a diameter of the largest droplets in the spray at the burner outlet. The Stokes number St, is the ratio of the drag relaxation time $\tau_p(S) = (\rho_l S)/(18\pi\mu_g)$ to the acoustic time τ_f, whereas the evaporation number E_v is taken to be the ratio of the evaporation time $\tau_v(S) = S/R_s$ over the acoustic time :

$$\mathrm{St} = \frac{\tau_p(S)}{\tau_f}, \qquad E_v = \frac{\tau_v(S)}{\tau_f}, \tag{5.2}$$

where ρ_l is the liquid density, μ_g the dynamic viscosity of the gas and R_s is a characteristic surface evaporation rate. Both numbers are linear functions of the droplet surface S.

Different liquids are used to vary the liquid evaporation rate. Decane is chosen as a reference slowly evaporating liquid, and methanol and heptane are more volatile. The mean size distribution for decane is presented in (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007) and is weakly polydisperse as the other distribution (see Table 5.1). The dynamical response is essentially the same for the three liquids but droplets of different size can have very different dynamics (see Stokes numbers in Table 5.1). Conversely, the evaporation numbers are very different from one liquid to another since droplets below 18μm diameter of heptane or methanol have evaporated within one gaseous acoustic time, i.e., one vortex eddy turn over time.

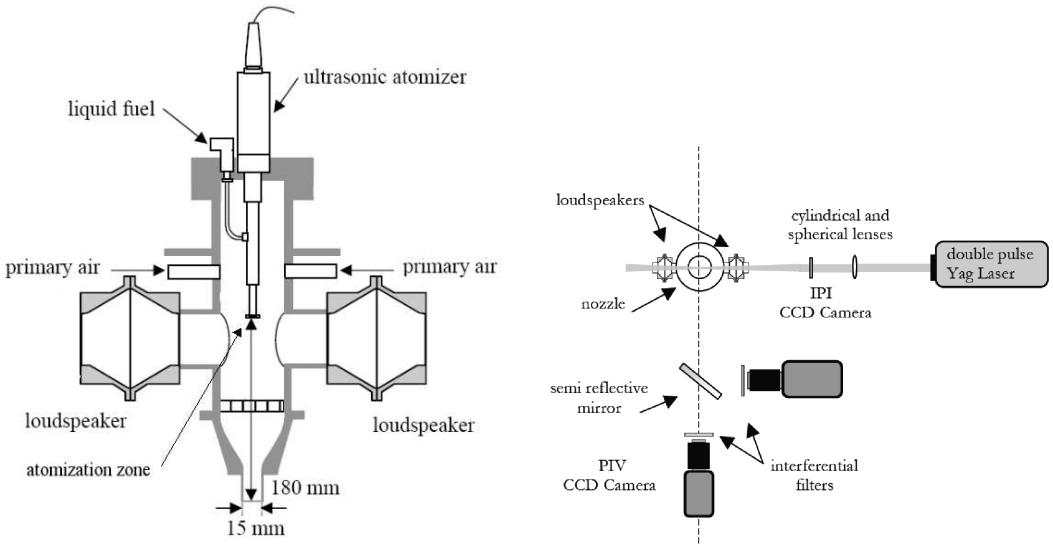


Figure 5.1 – *(left) Lateral view of the experimental setup, (right) PIV and IPI experimental set-up.*

5.2 Experimental devices and diagnostics

5.2.1 Experimental setup

The experimental facility is presented in Fig. 5.1. The injector is equipped with a converging nozzle with a symmetry of revolution leading to a flow at the nozzle exit with a turbulence level less than 1% of the mean axial velocity U_0=1.15 m.s^{-1}. A spray is generated from an ultrasonic atomizer, which is fed by a syringe (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007). The air jet flowing through this injector is acoustically forced by two loudspeakers placed on the sides of the facility to control the formation and evolution of vortices in the near field of the jet. Air flows through the cylinder into the facility and drags the droplets. In order to obtain a homogeneous flow at the nozzle exit, the distance between the atomizer zone, where volatile liquids start evaporating, and the nozzle exit is 180 mm and corresponds to a droplet residence time of a fraction of a second (Pichard, Durox, and Ducruix 2005). To benefit from the positive effect of gravity on the droplet formation and motion, the facility is put in the downward direction.

The modulation frequency is $f = 50$ Hz so that the Strouhal number is Sh = 1.0 in this study. At this Strouhal number the jet is yet highly sensitive and the excitation generates large periodic vortices closer to the nozzle exit, than for lower Strouhal numbers corresponding to the most amplified instability mode (Birbaud, Durox, Ducruix, and Candel 2007). High vorticity levels are reached in the core of the vortices (600 s^{-1}) (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007). The full cycle of oscillations is divided into 20 regularly spaced phases.

5.2.2 Measurement techniques PIV and IPI

The Particle Image Velocimetry (PIV) technique is used to characterize the gas flow and droplet velocities. The set-up consists in a double pulsed Continuum Nd :Yag laser. A laser sheet (500 μm thick) is focused on the central plane of the injector. The particle Mie scattering is collected on a Dantec Hi-Sense CCD camera 1600×1186 pixels^2 equipped with a Nikkor 60 mm lens and an interferential 532 nm filter. Velocity calculations are performed with an iterative cross-correlation algorithm (Dantec FlowManager). An iteratively decreasing interrogation window size is used with final value fixed to 16×32 pixels^2 with an overlap of 50 %.

The Interferometry Particle Imaging (IPI) technique aims at determining the size of transparent droplets, which flow through the light sheet and scatter light towards a camera. This technique is based on the interferences created in a voluntarily defocused image between reflected and refracted light rays, traveling through a transparent droplet. Using the geometrical approach (König, Anders, and Frohn 1986; Glover, Skippon, and Boyle 1995), the interference pattern between reflected and first order refracted light ray paths can be expressed by a simple linear relation between N_f the number of fringes and D the particle diameter

$$N_f = \kappa D, \tag{5.3}$$

where the geometrical factor κ is expressed by :

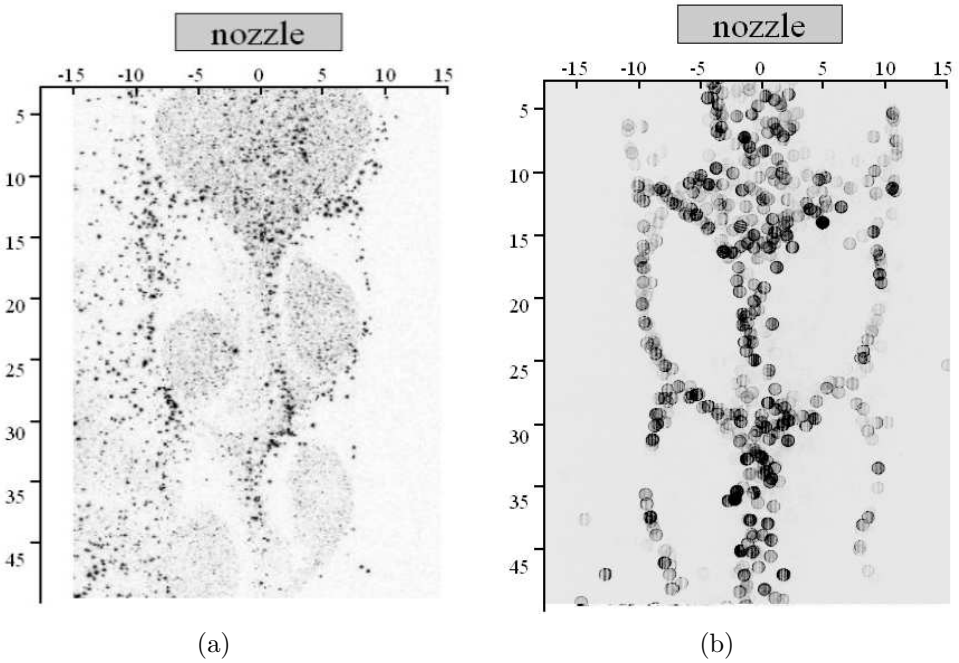
$$\kappa = \frac{\alpha}{\lambda}\left[\cos\left(\frac{\phi}{2}\right) + \frac{m\sin\left(\frac{\phi}{2}\right)}{\sqrt{m^2 - 2m\cos\left(\frac{\phi}{2}\right)+1}}\right] \tag{5.4}$$

λ is the light source wavelength, ϕ is the observation angle, relatively to the light source direction, m is the relative refraction index of the liquid related to the gas, α is the collection angle and can be expressed as:

$$\alpha = 2\sin^{-1}\left(\frac{d_a}{2z_l}\right) \tag{5.5}$$

where d_a is the lens aperture diameter and z_l the distance between the measurement plane and the camera lens plane. The present experiment is carried out with two CCD cameras collecting the scattered light at 90 deg relatively to the laser light source (see Fig. 5.1). The set up is constituted on the base of the PIV set up described in this paper, with two identical cameras. The PIV camera is not only used for the droplet velocity measurement but for the droplet position detection too and the second one is defocused to generate the interference fringes. Two different configurations have been used to achieve different objectives. Firstly, the global behavior of the excited jet is studied with a full view of the jet with at least three successive vortices (59×44 mm^2), with the same experimental set up as previously detailed in (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007), see Fig. 5.2. The maximum measurable droplet diameter by IPI with this set up is about

90 μm, as one fringe has to be 4 pixel wide at least to be detected, and the minimum diameter is 9.6 μm to have more than one fringe in the defocused circle. In order to obtain correct statistics, the image acquisition has been phase-locked by synchronizing the laser and the camera with the loudspeaker excitation signal. For each of the 20 phases, at least 20 phase-locked pairs of images are acquired and processed. Results are then phase-lock averaged. In the present contribution, we intend to provide droplet density results for both a non evaporating and an evaporating situations as well as size-conditioned droplet dynamics. The previous set up was not suitable for such investigations for two reasons: limited droplet diameter range and too low statistics. In order to reproduce the experiment with an evaporating liquid, the measurable diameter range should be shifted towards the smaller droplets. The distance z_l between the laser sheet and the the camera lens, is thus reduced to 180 mm (collection angle $\alpha = 6.8$ deg), leading to a smaller field of view (27×20 mm^2) and a better spatial resolution. The measurable diameter range becomes $[5.2, 97]$ μm. Moreover, the number of phase-locked acquisitions is increased to 120 in order to enhance the statistical resolution.



(a) (b)

Figure 5.2 – *Instantaneous modulated air flow and decane spray. (a) Tomographic focused image of the air flow seeded with oil (light gray particles) and the decane spray (black dots). (b) Defocused image of the decane spray. There is no oil seeding in the air flow.*

5.3 Numerical Approach

5.3.1 Numerical Resolution

A multi-fluid spray computation is done, to be compared to these experiments. We use drag and evaporation models with convective corrections and we take into account the gravity,

see Chpt. 2. This multi-fluid computation was done, with regard to transport, with a dimensional splitting for 2-D axisymmetric configuration, described in Sec. 3.2

5.3.2 Numerical setup

The numerical computation requires two types of input from experiments: 1- boundary condition for the dispersed phase spatially resolved in the radial direction at inlet for droplets mass densities and velocities in each section, 2- the full gaseous velocity field as a function of space and time to which the dispersed phase is coupled to in the numerical simulations.

For the latter, 20 phase-locked velocity fields obtained from PIV measurements are interpolated in time and space following the requirements of the dispersed phase discretization. After noticing in (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007) that the slight deviation of the gaseous flow field from the 2D-axisymmetry could lead to peculiar behavior of the spray along the centerline, and in order to be compatible with the spray description, the obtained velocity fields are symmetrized in a symmetry plane. The resulting modification of the gaseous flow field is of the amount of a few percent in L^2 norm and is mainly important far from injection.

The former spray injection boundary condition roughly corresponds to the nozzle exit. For each time step and in each size section (20 sections of 3 μm width in diameter covering the $[0, 60]\mu m$ diameter range), at the inlet axial position, we interpolate spatially the droplet velocities conditioned by size. The injection width of the spray strongly varies in time; at each time and in each section, the number density of droplet is taken to be constant as a function of radial position in agreement with experimental measurements.

In the present contribution, the spray volume fraction are bounded in areas of preferential concentration by 5.10^{-5} and is mainly much below such a value, thus justifying the use of one-way coupling; besides, up to measurement precision, the gaseous velocity field with and without the spray are shown to be the same. The computational domain is discretized in 20 sections in the size phase space, and in 400×400 structured cells in the two spatial directions, thus offering good resolution for the purpose of comparisons with experimental results. Eventually, such a simulation is a quasi three-dimensional (2D in space, 1D in size) unsteady simulation carried out during four periods in order to reach a periodic regime.

5.4 Results and discussion

5.4.1 Global spray behavior

Let us first describe the full view of the configuration which covers the evolution of three vortical structures at least issued from the nozzle exit. One vortex covers the 40mm axial distance in a time of about 80ms that is $4\tau_f$ and the eddy turn over time is about 20ms. If we first concentrate on decane, the evaporation rate is so low at the considered temperature, that the liquid can be considered to be non-evaporating. In Fig. 5.3, (left), it is observed from experimental measurements that droplets are essentially absent from the vortex cores, whereas the vortical structures create both successive zones of concentrations along the

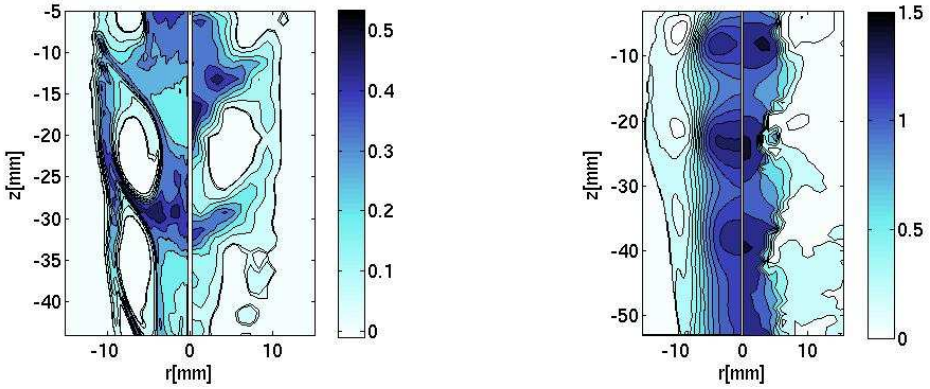


Figure 5.3 – *Comparison between experimental measurements (right half) and numerical simulation (left half) for decane droplet concentration (droplet/mm^3 - left picture) and mean droplet axial velocities ($m.s^{-1}$ - right picture) for the phase $\phi = 0$ (arbitrary reference). Experimental results are presented through an averaged field obtained from 20 instantaneous phase-locked samples for a 59×44 mm^2 field of view. Numerical simulation was obtained on a 400 × 400 grid.*

centerline and large arms preventing the captured droplets from flowing toward the exit of the domain. The absence of the spray from the vortex core is due to the combination of two phenomena: mixing of the spray with the air entrainment associated with the pulsation of the jet and ejection of droplets from the strong vortical structures due to inertia (see tomographic images of the pulsated spray in (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007)). The corresponding axial velocity of the droplets is presented in Fig. 5.3, (right). Let us underline that, extrapolating the results provided in (de Chaisemartin, Laurent, Massot, and Reveillon 2007), for the range of Stokes numbers associated with the biggest droplets under consideration, i.e. St = 0.43, an eddy turn over time is enough for an almost complete ejection of such droplets from the vortex present in our configuration, thus explaining the fact that droplets are being ejected after one vortex turn over as observed in the experiments. However, such an ejection of droplets from the vortex core is essentially governed by the size of the droplets and will be different depending on the size. Thus we need to focus on size-conditioned droplet dynamics. Before doing so, let us compare experimental measurements to numerical simulations at a global level.

First, the axial velocity map is presented for both experimental and numerical results in Fig. 5.3, (right). In the experimental case, velocities are measured essentially in the central region of the jet, as the droplets are absent in the vortex cores. Along the central axis, the droplets are successively accelerated and decelerated, corresponding to the vortex convection. The zones of maximum droplet velocities in the numerical velocity field are localized at the same position as in the experimental velocity field. Numerical results are thus able to reproduce the droplet axial velocity and such a good agreement is also obtained for radial velocity. Next, we compare droplet number density, Fig. 5.3, (left). It is measured from PIV results by considering the number of droplets in a given volume. The present results are representative of the general spray behavior. Concerning the numerical mean droplet concentration, high concentration zones are located at the head of each vortex. This result shows that the interaction between the high vorticity gas flow and the spray leads

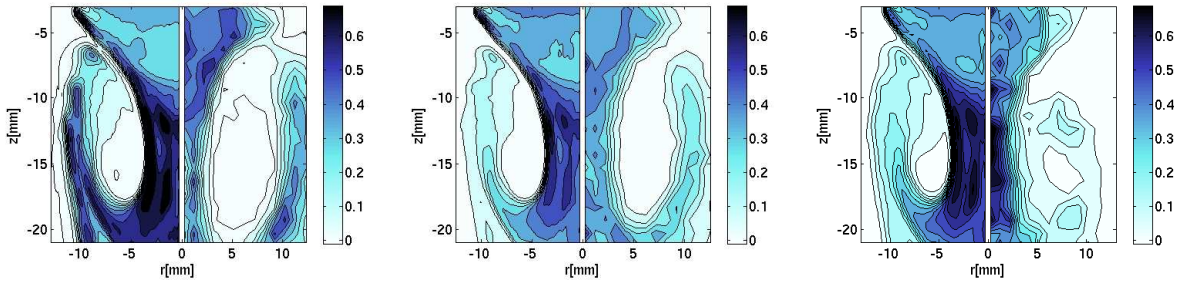


Figure 5.4 – *Comparison between IPI experimental measurements (right half) and numerical simulation (left half) for total droplet concentration at phase π in droplet/mm^3. From left to right: decane (non evaporating reference liquid), methanol and heptane (evaporating liquids). Experimental results are presented through an averaged field obtained from 120 instantaneous phase-locked samples with a reduced field of view of 27×20 mm^2. Numerical simulation was obtained on a 400 × 400 grid.*

to a non uniform distribution of droplets. The vortex cores present voids of decane droplet concentration whereas high droplet concentration appear at the leading edges (head) of the vortices. Droplets are ejected from the high vorticity core and follow the external vortex arms. Consequently, at a global level, the comparisons are very good. Let us underline the progress made in terms of comparisons in reference to (de Chaisemartin, Laurent, Fréret, Massot, Birbaud, Lacour, Ducruix, and Durox 2007); first, because of the coherence in terms of symmetry between the Eulerian spray modeling and the experimental gaseous velocity field, no peculiar behavior on the centerline is to be found. Besides, since we have included the variation of the spray width versus time at the inlet, and since we have greatly improved the experimental statistics, we reach a much more satisfactory level of comparisons. This level will be denoted "quasi-quantitative"; actually even if some differences appear in the comparisons and even if we can not claim a fully quantitative agreement, up to measurement uncertainty, we reach both the same structure and levels of droplet density and velocity.

5.4.2 Size-conditioned dynamics comparisons

In addition, one of the objectives of the present study is to put forward the pulsated jet behavior regarding its polydisperse nature. In order to vary the size distribution width and to have a larger range of droplet diameters in the jet, different liquids are studied on a reduced experimental view (27×20 mm^2). The experimental droplet concentration is obtained from 120 instantaneous image pairs and averaged in windows of 0.66 mm^2.

Fig. 5.4 presents the total concentration map obtained for decane (left), methanol (middle) and heptane (right) in the focused area under consideration. For the three liquids, droplets concentrate between the two vortices along the central jet axis and are ejected around the vortices to form external arms. The concentration reaches, in decane case, 0.4 droplet/mm^3 inside the arm which nearly catches up the central part of the jet. For the methanol spray, the pulsated jet structure is modified: the external arm concentration decreases to 0.2 droplet/mm^3. For the heptane spray, droplets are present inside the vortex core and concentration reaches 0.2 droplet/mm^3. The presence of evaporation thus changes the structure of the spray localization in both the vortex core and in the arms; such a

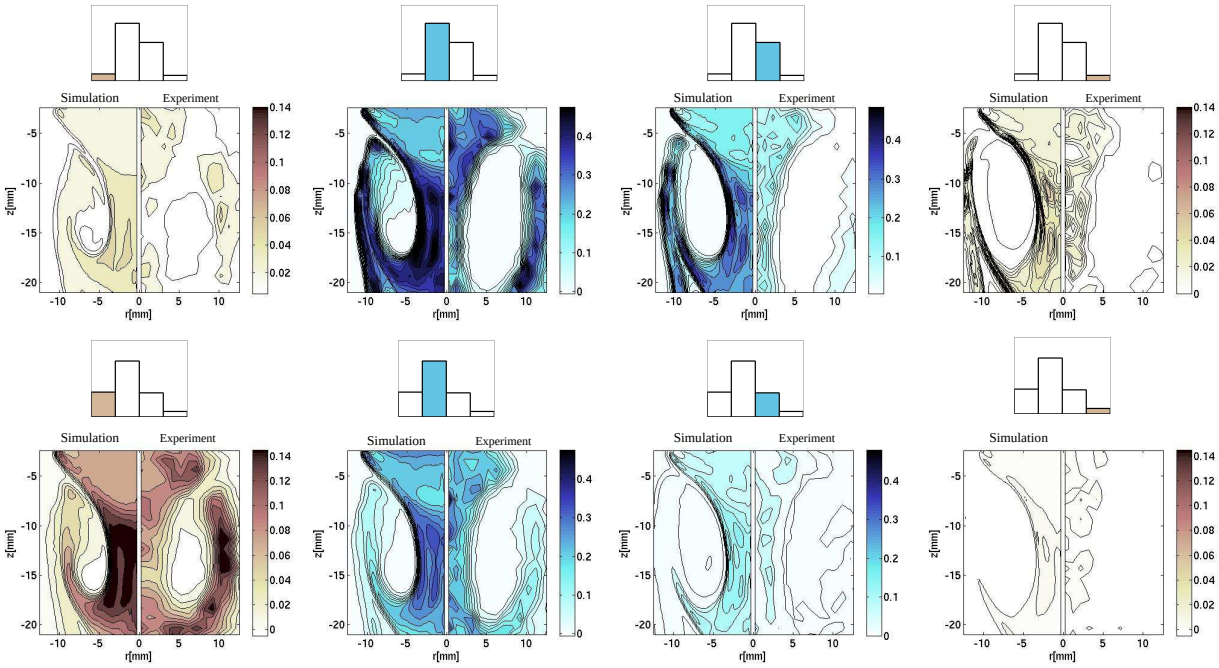


Figure 5.5 – *Comparison of size-conditioned decane (top) and methanol (bottom) droplet number density (droplet/mm^3) between 120 phase-locked IPI experimental measurements with a reduced field of view of 27×20 mm^2 (right half) and numerical simulations on a 400 × 400 grid (left half) at phase π. Four size ranges are presented from left to right, range [6,15] [15,30] [30,45] [45,60] in μm. Small histograms represent the droplet number repartition among the four size ranges. Two scales are used. Very small and big droplets have low concentration level and are presented in ochre, whereas the high concentration levels of droplets of inter size are presented in blue.*

phenomenon is well reproduced by the numerical simulations at a global level.

In order to further investigate the polydisperse nature of the spray and its evaporation, we focus on the first two liquids and investigate the dynamics conditioned by droplet size. Decane concentration maps corresponding to four size ranges are presented in Fig. 5.5, (top).

For bigger droplet classes, the experimental results are noisy due to a poorer statistic, since the droplet number density is rather low. However, it is clear from experiments that the ejection of the droplets from the vortex core follow different dynamics depending on the inertia of the droplets. As size increases, the zone without droplets widens. Such a behavior is well captured by the numerical simulations, where the structure of the arm is found to be very different depending on the droplet size, thus exhibiting differential dynamics conditioned by size.

In the evaporating configuration, methanol concentration maps presented with the same four size intervals are presented in Fig. 5.5, (bottom). The differential dynamics observed in the previous case is maintained in such a case, even if the repartition of droplet number density is very different in terms of level due to evaporation. Let us underline that droplets

in the arms have travelled about along one eddy along its turn over, thus corresponding to a time of about $2\tau_f$ and showing an important role of evaporation. However, evaporation yields additional differences. We can note a large amount of small droplets due to the evaporation phenomena and as the size increases, even if the area without droplet is enlarged, the structure of the created arm is rather different since the small droplets present in this arm have had various dynamics during the evaporation history and followed different paths from the small decane droplet. Such a phenomenon is captured by the numerical simulation, however, the statistics level is not high enough to capture it in the experiments. Once again, experimental and numerical results present very good agreement, thus demonstrating the ability of the Eulerian model to capture the differential dynamics of droplets of various size in a Eulerian manner. Let us also underline the robustness of the numerical methods used for the simulations since we were able to conduct a numerical simulation of the disperse liquid phase coupled to the gaseous flow fields obtained from experimental measurements.

In the present configuration, the spray dynamics leads to areas of preferential concentration of droplets which can eventually lead to a limitation of the IPI technique. Indeed IPI results rely on the processing of images and fringes counting, thus leading to a validation rate which can spatially vary depending on the droplet concentration. However, we have checked the total number of droplets present in the window of comparison is the same between experiments and numerical simulation, thus showing that the potential bias associated to higher droplet concentration does not have an important impact on the experimental measurement in these areas.

Eventually, let us emphasize that this contribution provides the first quasi-quantitative comparison of size-conditioned dynamics between experimental measurements and numerical simulation. This will be essential for more realistic configurations and for combustion chambers.

Conclusion

This study completes multi-fluid model characterization, complementing the mathematical and numerical studies of Chpt. 3 and Chpt. 4 by experimental assessment. It provides innovative experimental techniques allowing to capture quantitatively the size/velocity correlations of the injected spray. The computations realized show again the robustness of the numerical method provided, allowing to take the experimental data fields to describe the gas flow in the computation. Finally, these comparisons validate the precision of the multi-fluid model and its ability to capture the dynamics of the spray ejection by the gaseous vortices. This study validates the behavior of the multi-fluid model in describing non-evaporating and evaporating spray, through the use of different fuels in the experiments. This study represents a first step toward simulation/experiment confrontations for two-phase combustion.

Part III

Scientific computing: genericity and efficiency in multi-fluid framework

Introduction

The latest advances in High Performance Computing (HPC), made possible the predictive simulation of an industrial burner, through the use of massively parallel computing. This type of simulations is associated to very efficient computational codes, as for instance the AVBP solver co-developed at CERFACS and IFP. This type of predictive computation is, at present, restricted to purely gaseous combustion cases. Indeed, they are still several open questions concerning liquid injection computation. Academical studies are thus needed in this framework. Nevertheless, given the final objective, that is industrial device predictive simulation, the academical studies must tackle operational concerns. Indeed, the computational efficiency of the developed method is of crucial importance, and must be associated with the modeling concerns. In this context, academical solvers have to be developed, fulfilling two main constraints:

- provide a high genericity level, to concentrate a wide ranges of models, and to extend easily to new outbreaking methods;
- provide relevant conclusions concerning the use of the assessed methods in a HPC context.

The study of the multi-fluid method provided in this work, is included in the actual academical research on spray resolution. A key element of the present study, is to provide an academical solver allowing to meet the objective of multi-fluid operational evaluation. The final objective being to show multi-fluid relevance for industrial device simulation. This academical solver must conciliate a high genericity level together with an important computational efficiency.
This part provides:

- an original answer to genericity requirement, making the most of Fortran standard, through object oriented emulation and intrinsic scientific computing-devoted functions use, Chpt. 6;
- an original optimization coupling single processor optimization and parallel domain decomposition in a 4-D framework, Chpt. 7.

This part shows the features of the developed solver, MUlti-fluid Solver for Eulerian Spray (MUSES3D), detailing the algorithms and the programming language implementations provided to ensure genericity. The resulting genericity allows to couple the multi-fluid method to a gas solver and to tackle various configurations, Chpt. 9, Chpt. 10 and Chpt. 11. Besides, it enables the efficient implementation of new types of multi-fluid methods. Indeed, high order moment methods for evaporation have been included, (Massot, Laurent, Kah, and de Chaisemartin 2009). Furthermore, the recent multi-fluid method for out of equilibrium velocity distributions has been assessed in jet configurations, (de Chaisemartin,

Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009; Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009), using the solver.
Moreover, this part provides the details of an efficient parallel multi-fluid implementation, from the decomposition strategy analysis, to the types of communications used. It allows to assess the efficiency of the multi-fluid method in a parallel framework. This first step is needed before tackling 3-D configurations, Chpt. 11, that are essential for an operational evaluation of the method.

Chapter 6

Research solver development: genericity requisite

Contents

Developing academical numerical tools allows to conduct efficiently the characterization of emergent methods prior to their use in industrial applications. Nevertheless, scientific computing evolution in the last years must impact academical tool development. We assess in this chapter the question of the required genericity in scientific computing developments. For an academical research solver, genericity appears as an essential issue. Indeed, a such solver must concentrate various developments of the research field considered, and evolve rapidly in order to introduce the new developments. It must thus have a modular structure, built up on different blocks sharing services.

In the framework of the multi-fluid model, we expect an efficient genericity at three levels. First, different Eulerian spray resolution methods have to be used. Indeed, high order methods for evaporation or multi-velocity methods for non-equilibrium velocity distributions, Chpt. 2, have to be implemented, in addition to the "classical" multi-fluid method assessed in this study. Implementing these different methods in a solver is not in itself a major difficulty, nevertheless making these models share the most elements as possible, in order to obtain a high efficiency for their implementation and evolution is a more challenging issue. Second, the different modeling frameworks introduced in Chpt. 1, are needed. The simplified framework is needed to conduct the refined studies of Part. II, or to evaluate the multi-fluid method in more complex flows, Part. IV. However, a more general framework must be implemented, not to be restricted to these needed simplifications. Third, in order to assess more realistic multi-fluid computations, 2-D axisymmetric and 2-D plane jets and 3-D isotropic turbulence are achieved in Part. IV. It requires to handle efficiently various types of boundary and initial conditions.

A key issue arising in this context concerns the choice of the programming language. Due to the recent and planned evolutions of the FORTRAN standard, and given the various intrinsic, scientific computing devoted, functions it holds, (Metcalf, Reid, and Cohen 2004; Corde and Delouis 2008), this standard has been chosen. The 90/95 standard, actually supported by compilers, allows to emulate object oriented programming, (Decyk, Norton, and Szymanski 1997). Furthermore, very efficient array manipulation functions are available, (Corde, Delouis, and Dupays 2008). An original position is thus adopted, with the development of a solver making the most of Fortran actual standard, for genericity purpose. This chapter presents the main choices done in this context for the development of the numerical solver for multi-fluid description, MUlti-fluid Solver for Eulerian Spray (MUSES3D). The idea through this presentation, is to highlight the scientific computing issues related to a such development, and to show the choices made for their resolution. In this framework, the different blocks devoted to data structures, numerical schemes and boundary conditions are described. The genericity provided in the development, allows an easy coupling with a gas phase solver.

6.1 Research solver and scientific computing issues

The main scientific computing issues tackled within this chapter are detailed in this section. These issues are related with the expected applications one wants to reach with the solver MUSES3D developed in this work for multi-fluid spray computations. We show the relation between the structure of the multi-fluid solver developed and the scientific computational issues that are addressed in details in Sec. 6.2 to Sec. 6.5. A key question addressed in this section is the choice of the development language. We aim at using a language devoted to scientific computing and albeit allowing a high level of genericity.

6.1.1 Scientific computing issues raised in MUSES3D

One intends to use the multi-fluid solver in various cases. We aim at using:

- various computational domains. Indeed 2-D, 2-D axisymmetric and 3-D cases are aimed at, see Part. IV. This requirement has an influence on the whole solver structure: data types, numerical schemes, boundary conditions, etc.
- Different configurations are also expected. Indeed, jet with spay injection, Chpt. 9 and Chpt. 10, as well as droplet-laden vortical flows, Chpt. 11, are realized. Therefore different types of boundary conditions are considered:
  - periodic,
  - injection,
  - free outlet,
  - symmetry.
- Different types of resolution methods have to be implemented. Indeed, although devoted to multi-fluid spray resolution, the solver should provide different multi-fluid variants, described in Chpt. 2. High-order methods for velocity and size phase space are implemented. The number of equations and variables change from one method to another, as different sets of moments of the spray NDF have to be considered, Chpt. 2.
- Different numerical schemes have to be provided by the solver. The first reason is to have for the same multi-fluid method different possibilities. For example, the dimensional splitting algorithm may take different forms, see Chpt. 3. Furthermore, albeit dimensional splitting clearly represents the most efficient way to treat the configurations computed in Part. IV, the extension to multi-dimensional schemes has to be designed in the solver. Second, the different types of multi-fluid methods, require different resolution schemes.
- Different types of droplet models are also expected. Indeed, the evaluation of resolution methods, as provided in this work, requires a simple modeling context. Nevertheless, the use of multi-fluid resolution with more precise modeling, as infinite liquid conductivity model, will have to be done later on. The number of variables of the problem may then change, depending whether heat transfer are resolved or not.

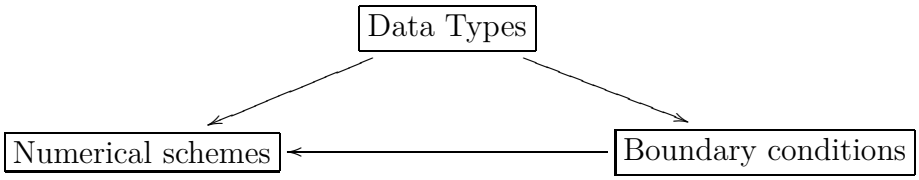


Figure 6.1 – *Modular block structure of MUSES3D solver*

A high level of genericity is therefore needed for MUSES3D, in order to treat all these cases. Furthermore, it has to be coupled with a high modularity, in order to be easily coupled with a gas phase solver, as detailed in Sec. 6.5. These requirements are impacting the solver at different levels. The main blocks of MUSES3D solver are summarized on Fig. 6.1. Each block requirements lead to difficulties.

- The data types defined in the solver must handle any dimension number for the computational domain. Furthermore, the number of variables solved varies due to the number of spray NDF moments solved or to the types of droplet models used. One thus wishes to have generic data types as well as generic manipulation procedures to handle the different possibilities and to extend to new cases easily. It corresponds to an object-oriented class definition, gathering data structures and methods for their manipulation. The strategy chosen in MUSES3D is detailed in Sec. 6.2.
- The numerical schemes used for the multi-fluid computation use two types of solver, for the physical space and for the phase space, driven by a splitting algorithm. Given the requirements of genericity for the numerical schemes, we provide in MUSES3D a generic solution for splitting algorithms implementation, a general finite volume transport scheme implementation, and an efficient coupling with an ODE solver. These elements are described in Sec. 6.3.
- Finally, boundary conditions treatment must be isolated from numerical scheme implementation to give the expected modularity. We provide in MUSES3D an original array manipulation for the boundary conditions treatment, detailed in Sec. 6.4.

These scientific computing issues are arising for the development of the MUSES3D solver. They are illustrated in this framework in Sec. 6.2 to Sec. 6.5. Nevertheless, one can note that these issues are quite general and that the solutions provided in MUSES3D implementation could be used in another framework.

6.1.2 Language choice for formula translation

The genericity we wish to reach would ask for a totally object oriented language. In this framework C++ would be a very good candidate. This is confirmed in (Cary, Shasharina, Cummings, Reynders, and Hinker 1997) where C++ and Fortran90 are compared for object-oriented scientific programming. Indeed, C++ is a full-featured object-oriented programming language, while Fortran90 allows object-oriented programming only to some degree. Furthermore, as pointed out in (Cary, Shasharina, Cummings, Reynders, and Hinker 1997), in the general software engineering community, C++ is order of magnitude more common than Fortran. Nevertheless, this is not the case in the scientific computing community where Fortran programming language is still widely used. Furthermore,

though its evolution was quite slow, the recent and planned changes make it a still very competitive high level language completely devoted to scientific computing. Indeed, the 90-95 standards, today completely supported by compilers, introduce many features allowing a high level of genericity and a highly modular implementation. This can be achieved through the combined use of `type` and `module` syntax elements. Furthermore these norms introduce many array intrinsic functions devoted to scientific computing, allowing easy and efficient array transformations as well as vector or matrix computations, see (Corde, Delouis, and Dupays 2008) for example. Moreover, as said in (Cary, Shasharina, Cummings, Reynders, and Hinker 1997), Fortran compilers may be more optimized on massively parallels processors. Finally, the future of Fortran, defined in the 2003 standard, is, among many improvements, full object orientation as well as C interoperability, see (Corde and Delouis 2008) and (Metcalf, Reid, and Cohen 2004). This norm is at present being implemented on compilers, and two complete implementations are already available. We thus use Fortran programming language in this work. We insist on the fact that, even if Fortran is the most common language for scientific computing in aeronautics and automotive applications, our approach is original. Indeed, we want to make the most of Fortran object-oriented programming support and scientific computing devoted features. This approach corresponds to a first step toward the use of the Fortran 2003 standard, that is a full object-oriented programming language devoted to scientific computing.

Conventions for algorithm and code writing

Throughout this part, we will use two ways to write implementation solutions: if the feature illustrated is independent of any language consideration, we will write the algorithm with the convention of Algo. 3. On the other hand if the underlined algorithm is based on specific Fortran features, as for example intrinsic procedures, we will write the Fortran code translation of the algorithm, in typewriter font, as done in Fort. 1. Therefore,

Algorithm 3 Algorithm nomenclature

```
{Algorithm Beginning}
begin block constructs
   instructions
   variable := value {Assignment}
end block constructs
...
call myprocedure(arguments)
...
{Algorithm End}
...
{Procedure definition}
procedure myprocedure(arguments)
   procedure content
end procedure myprocedure
```

two types of fonts will be used when referring to a procedure, whether its algorithm or Fortran implementation, is given. If the algorithm is given, we refer to myprocedure, whereas if the Fortran implementation is given, we refer to `myprocedure`. In both algorithm

Fortran 1 Fortran code nomenclature

```
!--Code beginning
...
begin block constructs
  instructions
end block constructs
...
call myprocedure(arguments)
...
!--Code end
...
!--Procedure definition
subroutine myprocedure(arguments)
  procedure content
end subroutine myprocedure
```

and Fortran code, we will use arrays of variables. We will define arrays in the following way:

myarray $(N1, N2, N3, N4, N5)$,

where myarray is the name of the array and $(N1, N2, N3, N4, N5)$ its shape. The shape corresponds to the sequence of extents, where the extent is the number of elements along a given dimension. The rank of the array is the number of dimensions, e.g. five for the example above.

Fortran genericity and modularity for scientific computing

Fortran90/95 standards provide several features devoted to genericity, modularity for scientific computing applications. The main feature used for the development of MUSES3D are summarized here.

- Regarding genericity, Fortran gives the possibility to define new **derived data types**. Different intrinsic types can then be gathered in an object. For example one could gather droplet characteristic in an intrinsic type `droplet_type`, Fort. 2. The decla-

Fortran 2 Fortran derived data types illustration

```
type droplet_type
  real  ::  position, diameter, Temperature
  real, dimension(3)  ::  velocity
end type droplet_type
```

ration statement for an element of this type would be:

```
type (droplet_type) :: dropleta, dropletb
```

The components of this variable are addressed individually using the component selector character percent (%). Furthermore, Fotran90/95 standards offer **operator overloading** and **generic interfaces**. Therefore, operations can be defined for the

derived data types created. One could for example define assignment, overloading assignment operator =, and give a meaning to:

```
dropleta = dropletb
```

Besides, generic interfaces allow to call several procedures by the same generic name. It thus gives the possibility to define high level function used by the developer, regardless of the types of the variables considered. For example an `allocation` generic interface could be defined to allocate in the same way both arrays and derived types containing arrays. Derived types associated to generic interfaces allow an emulation of object orientation, defining classes gathering data types, public or private, and methods, i.e. procedures associated to these data types.

- The required modularity for MUSES3D, can be obtained using Fortran **modules**. They allow a very efficient sharing of data and procedures. Besides it allows to define easily explicit interfaces for procedures. In this case, the interface of the procedure, i.e., the name and properties of the arguments, is known by the calling unit. It leads to a safer programming. Finally, a module can declare its content to be `private`, allowing data encapsulation. Associated to the `type` feature, it allows to define an abstract object.
- Finally, we take advantage of the several **intrinsic array manipulation functions** defined in Fortran90/95. Among these functions, array shifting functions are very interesting for boundary conditions implantation. For example, the function `eoshift(array,shift[,boundary][,dim])`, shifts every rank one section of `array` across dimension `dim` circularly and inserts values `boundary` into the gaps so created. This function will be detailed later on, as it was used to pilot MUSES3D boundary conditions, see Sec. 6.4.

Moreover, as far as precision is concerned, the kind type parameters are used in order to work with the same precision on any computers. It allows to set the number of Bytes used to describe numbers or characters. For example, to encode a real `x`, on height Bytes and an integer `n`, on four Bytes, one writes:

```
integer           ::  ip=4, rp=8
real(kind=rp)     ::  x
integer(kind=ip)  ::  n
```

Therefore, to set the precision of a constant `cst`, one can write `cst_ip` or `cst_rp` for an integer or a real constant.

6.2 Derived data types

6.2.1 Data structure

The ability to implement easily various spray model lies in the efficiency of the data structure for the spray variables. This structure gathers Eulerian spray data as mass, momentum or more generally moments of the spray NDF, and possibly abscissas and weights when using more complex methods based on quadrature of moments, as needed in the multi-fluid

multi-velocity model, see Chpt. 2. The multi-fluid models can need size moments, velocity moments and thermodynamical temperature moments. We thus define, in Algo. 4 the spray structure, with five arrays for, respectively, velocity moments, size moments, temperature moments, abscissas and weights, Algo. 4. The mass density is a zero-order moment in

Algorithm 4 Spray data structure

```
structure spray{Spray structure}
  array velmom, sizemom, tempmom, abs, wgt
end structure spray
```

size and velocity and can then be included in velmom or sizemom. These spray variables, velmom, sizemom, etc., are chosen with a rank equal to five, denoting all these variables by the generic name spray_variable, we have:

spray_array $(N_x, N_y, N_z, N_S, N_{\text{mom}})$

where the shape (N_x, N_y, N_z) covers the physical domain, N_S stands for the number of multi-fluid size sections and N_{mom} the number of considered moments. Although arrays with a high rank are not very well suited for computational efficiency, this choice is the only way to treat dynamically the number of sections and the number of moments considered, a mandatory feature for multi-fluid computations. This data structure is defined for the model variables but will be shared to define other elements, see Algo. 5. First, the physical transport numerical scheme will use this structure. Indeed we need to define numerical fluxes and possibly slopes and reconstructions in second-order method, see Sec. 3.2. These scheme variables need to be defined for all the spray variables and we thus define new structures for the scheme variables inheriting the spray structure. We define for example the numerical flux structure, first structure of Algo. 5, as well as slope and reconstruction structure in the same way. Nevertheless, for these numerical scheme structure, the arrays

Algorithm 5 Composed data structures

```
structure numerical flux{Numerical scheme structure}
  structure spray
end structure numerical flux

structure boundary condition{Boundary condition structure}
  structure spray
end structure boundary condition
```

velmom, sizemom, etc., are not with a rank equal to five, but to six. We denote these variables under the generic name num_array:

num_array $(N_x, N_y, N_z, N_S, N_{\text{mom}}, N)$

For the flux variable, $N = N_{dim\_transp}$, i.e., the number of dimension of the domain, N_{dim}, if we use a fully multi-dimensional scheme, or one if we use dimensional splitting. This array dimension allows to store generically the flux components for each direction and to support 1-D to 3-D numerical schemes. We also have $N = N_{dim\_transp}$ for the slope variables, to store the slope for each direction when using a multi-dimensional scheme. Finally, for the reconstruction variables, $N = 2N_{dim\_transp}$ and represents the number of reconstructions

of the scheme, see Sec. 3.2 for details on the reconstruction procedures. We sum up the conditional values of this two variables, N and N_{dim_transp} in Algo. 6.

Algorithm 6 Value for variable N and N_{dim_transp}

```
{6-D Arrays last direction}
if (var = flux_var or var = slope_var) then {N ⇒ number of transport dimension}
   N := N_dim_transp
else if (var = reconstruction_var) then {N ⇒ number of scheme reconstructions}
      N := 2N_dim_transp
end if
...
{Number of transport directions}
if (dimensional splitting) then
   N_dim_transp := 1
else
   N_dim_transp := N_dim
end if
```

Second, we need to define boundary conditions, and store associated ghost cells, for all the spray variables and we thus inherit again the spray structure, see second structure of Algo. 5. Once again, the arrays involved in this structure will differ from the original spray array: indeed we cover only $N_{\text{dim}} - 1$ space dimension on the boundary. The boundary structure will be derived for the spray variables as well as for numerical scheme ones. We thus define two types of arrays for ghost cells storage:

bc_spray_array $(N1, N2, N_S, N_{\text{mom}})$, bc_num_array $(N1, N2, N_S, N_{\text{mom}}, N)$

$(N1, N2)$ being (N_x, N_y) for the third-direction boundary, (N_x, N_z) or (N_y, N_z) for the second- and first-direction boundary, respectively.

These data structures are implemented in Fortran through derived types. Such types allow to gather in a structure, considered as a scalar, heterogeneous elements. We have seen previously that all the structures use the set of variables of the model but in different shape arrays. We therefore defined three derived types, one for each rank needed, written in Fort. 3. One can note that these fields use dynamic memory allocation for their components, that is a feature of Fortran 2003 standard; nevertheless this standard is today supported in many compilers. If not, it is possible to use the Fortran 90 pointer type for the arrays, that is less suited for computational efficiency. The array `N(i,j)` is used for the allocation procedure. The shapes of the type arrays are stored in it. The shape of `N` is : `N`(number_of_arrays, 2× rank_of_arrays), to store the first and last index of each direction for the type arrays. These types allow to define the structures needed for spray, numerical scheme and boundary condition variables:

```
type(field_5d)  ::  spray
type(field_4d)  ::  Bound_spray
type(field_6d)  ::  Recons, Slope, Flux
type(field_5d)  ::  Bound_Recons, Bound_Slope, Bound_Flux
```

Fortran 3 Spray derived data types

```
type field_6d
  real(kind=rp), allocatable, dimension(:,:,:,:,:,:)  ::  sizes, vel, &
                                                          sizes_vel, temptherm
  integer(kind=ip), dimension(4_ip,12_ip) :: N
end type field_6d
!
type field_5d
  real(kind=rp), allocatable, dimension(:,:,:,:,:)  ::  sizes, vel, &
                                                        sizes_vel, temptherm
  real(kind=rp), allocatable, dimension(:,:,:,:,:)  ::  a, w
  integer(kind=ip),dimension(6_ip,10_ip) ::  N
end type field_5d
!
type field_4d
  real(kind=rp), allocatable, dimension(:,:,:,:)  ::  sizes, vel,
                                                      sizes_vel, temptherm
  real(kind=rp), allocatable, dimension(:,:,:,:)  ::  a, w
  integer(kind=ip),dimension(6_ip,8_ip) ::  N
end type field_4d
```

6.2.2 Class organization

In order to handle these structures, we define dedicated procedures completing:

- structure allocation/deallocation,
- structure arithmetical operations,
- structure read/write (I/O) operations,
- structure initialization.

We define for example a procedure allocate, allowing to allocate dynamically memory to the different structures with the same call:

call allocate(spray)
call allocate(flux)

We also want to be able to perform arithmetical operations directly on the structures:

$spray_3 := spray_1 + spray_2$
$flux_3 := flux_1 + flux_2$

To define such procedures, Fortran gives two features :

- generic interfaces,
- operator overloading.

We illustrate first the use of generic interface to generate a generic allocate procedure. To do so we define three procedures to realize allocation of the types `field_xd`, the example of `x=5`

is written in Fort. 4. This subroutine is based on the subroutine `fill_N`, devoted to write

Fortran 4 5-D field allocation
```
subroutine alloc_field5d(Field,Nd2)
type(field_5d), intent(inout) ::  Field
integer(kind=ip), intent(in)  ::  Nd2
!
call fill_N(Field%N,Nd2)
call allocate_array(Field)
!
end subroutine alloc_field5d
```

in `N` the shapes of the arrays. For the arrays of the numerical scheme num_array we need to know the value of N. This is done with the dummy argument `Nd2`. For the spray_array or the bc_spray_array we set `Nd2=0`. The rest of the profile $(N_x, N_y, N_z, N_S, N_{\text{mom}})$ is known from the data structure of the mesh, globally available in the solver. We finally write a generic interface, Fort. 5, to define a single allocation procedure `allocate`. An allocation

Fortran 5 Allocate generic interface
```
interface allocate
  module procedure alloc_field6d, alloc_field5d, alloc_field4d
end interface allocate
```

of a spray and a flux structure would then be:

```
call allocate(spray,Ndim_transp)
call allocate(Flux,Ndim_transp)
```

An extra difficulty arise for boundary condition structure allocation since we don't know a priori which bound we are treating and thus the pair $(N1, N2)$. We explain in details the boundary conditions treatment in the next paragraph.
Second, we illustrate the operator overloading defining the addition operator for two structures. We show here the example of a `field_5d` type. We first write in Fort. 6, the function defining the operation. The operator `(+)` is then overloaded with the interface block, Fort. 7 and will use the module procedure matching the type of the operands.

6.2.3 Composed-class for boundary condition treatment

In order to be able to treat 1-D to 3-D configurations, we need to define dynamically the number of boundary conditions, equal to twice the number of dimensions of the computational domain. We thus would like to have a dynamic array of boundary condition structure, `field_4d`, to store the ghost cells at each frontier. We define, in Fort. 8, the structure `comp_field_4d`, realizing this feature. It consists of a dynamic array of `field_4d` type elements, and a scalar `Ncomp`, giving the length of `c`, i.e., the number of boundary conditions to treat in the computational domain. In order to treat boundary conditions for the numerical scheme variables, we construct in the same way a structure `comp_field_5d` based on the type `field_5d`. In order to obtain methods to handle these new data types, we extend the ones developed for the basis data types `field_xd`. The only difficulty is for the allocations

Fortran 6 Addition operator overloading function

```
function add_field5d(fielda, fieldb)
type(field_5d), intent(in)  ::  fielda, fieldb
type(field_5d)  ::  add_field5d
!
!-Sizes moments
add_field5d%sizes = fielda%sizes+fieldb%sizes
!-Velocity moments
add_field5d%vel = fielda%vel+fieldb%vel
!-Size-Velocity moments
add_field5d%sizes_vel = fielda%sizes_vel+fieldb%sizes_vel
!-Thermodynamic temperature moments
add_field5d%temptherm = fielda%temptherm+fieldb%temptherm
!-Abscissas & weights for quadrature-based multi-fluid multi-velocity method
if (multi_v) then
  add_field5d%a = fielda%a+fieldb%a
  add_field5d%w = fielda%w+fieldb%w
endif
end function add_field5d
```

Fortran 7 Generic interface for addition overloading

```
interface operator (+)
  module procedure add_field6d, add_field5d, add_field4d
end interface
```

Fortran 8 4-D composed derived data type definition

```
type comp_field4d
  type(field_4d), dimension(:), allocatable  ::  c
  integer(kind=ip)  ::  Ncomp
end type comp_field4d
```

of the array components of these new types `comp_field_xd`. Indeed, are these arrays are storing the ghost cells at the boundaries of the computational domain, their shapes depend on the considered boundary. For example, in a 3D-case, the boundaries and the shapes of their associated ghost cells is defined as follow:

$bound_1 \iff x$-left boundary $(x = 0)$ => shape: gc1_array(N_y, N_z)
$bound_2 \iff x$-right boundary $(x = N_x)$ => shape: gc2_array(N_y, N_z)
$bound_3 \iff y$-left boundary $(y = 0)$ => shape: gc3_array(N_x, N_z)
$bound_4 \iff y$-right boundary $(y = N_y)$ => shape: gc4_array(N_y, N_z)
$bound_5 \iff z$-left boundary $(z = 0)$ => shape: gc5_array(N_x, N_y)
$bound_6 \iff z$-right boundary $(z = N_z)$ => shape: gc6_array(N_x, N_y)

We thus define an array compval containing the shapes of each boundary of the domain, i.e., for the boundary i, the ghost cells array gci_array has the following shape, regarding physical space:

compval(i,:) $= (N1_i, N2_i)$

We define a first level procedure, Fort. 9, devoted to boundary condition, to fill the array `compval` and to launch a generic procedure for `comp_fieldxd` type allocation, extended from `field_xd` type allocation procedure. Here for the example of the `field_4d`.

Fortran 9 4-D composed derived data type allocation

```
subroutine alloc_bound4d(Bound,Nd2)
type(comp_field4d),  intent(inout)   ::  Bound
integer(kind=ip),  intent(in)        ::  Nd2
!-local variable
integer(kind=ip), dimension(:,:), allocatable  ::  comp_val
allocate(comp_val(Bound%Ncomp,4_ip))
call fill_compval(comp_val)
allocate(Bound%c(Bound%Ncomp))
call allocate_comp_field(Bound,comp_val,Nd2)
!
end subroutine alloc_bound4d
```

6.3 Modularity for numerical scheme implementation

The core of the solver development is the ability to make the numerical scheme evolve as easily as possible. A change in the numerical scheme can, for our multi-fluid solver, affect the splitting algorithm as well as the physical transport scheme or the phase space one. These three entities must then be developed separately and be as modular as possible.

6.3.1 A generic splitting algorithm

The numerical method used for multi-fluid simulations is based on an operator splitting, see Sec. 3.2.1. As we mentioned in that section, different splitting algorithms are possible. Indeed, two types of Strang algorithm can be defined, based on a splitting time step equal

to the solver timestep of twice the solver timestep, see Sec. 3.2.1. In the following, we will refer to the Strang splitting based on the timestep Δt , as "classical" Strang splitting, and to the one based on timestep $2\Delta t$, as "modified" Strang splitting. Moreover, we can define a simpler splitting algorithm ensuring time first order, the Lie splitting algorithm, by simply solving separately physical space transport and phase space transport over Δt:

- Phase space transport: solve Eq. (3.12) during Δt
- Physical space transport: solve Eq. (3.13) during Δt

In order to assess the influence of these different algorithms on the simulations, we want to be able to switch easily in the code from one to another. Furthermore, we want to implement easily new splitting algorithms. We thus decide to implement operator splitting the following way:

Algorithm 7 Splitting algorithm

for i=1 to N_{step} **do**
  call phys_tranp($dt_{\text{phys}}(i)$)
  call phase_tranp($dt_{\text{phase}}(i)$)
end for

The variable N_{step} represents the number of steps of the splitting algorithm, i.e., two for Lie and modified Strang splitting, and three for the "classical" Strang splitting. The arrays $dt_{\text{phys}}(:)$ and $dt_{\text{phase}}(:)$, contain the timestep needed for each solver: $\Delta t/2$, Δt or zero. This zero-timestep leads to the immediate exit of the phys_transp or phase_transp solvers. For a "classical" Strang splitting we have:

$N_{\text{step}} := 3$
$dt_{\text{phys}}(1 : N_{\text{step}}) := (0, 1, 0)$
$dt_{\text{phase}}(1 : N_{\text{step}}) := (0.5, 0, 0.5)$

This case is equivalent to the first splitting algorithm described at the beginning of part Sec. 3.2.1. If we want to invert the position of the physical and phase space transports, we just need to change their timestep array. For the "modified" Strang splitting we need to change at each iteration these timestep array to alternate the order of the splitting steps:

Algorithm 8 Modified Strang Splitting timestep

if (odd iteration) **then**
    $dt_{\text{phys}}(1 : N_{\text{step}}) := (0, 1)$
    $dt_{\text{phase}}(1 : N_{\text{step}}) := (1, 0)$
else
    $dt_{\text{phys}}(1 : N_{\text{step}}) := (1, 0)$
    $dt_{\text{phase}}(1 : N_{\text{step}}) := (0, 1)$
end if

This implementation allows a high flexibility for the operator splitting definition.

6.3.2 Finite volume solver

Objectives

Regarding the physical space transport scheme, detailed in Sec. 3.2.2, we want to implement a general modular structure devoted to finite volume schemes. Such a scheme can be written, as mentioned in Sec. 3.2.2, the following way, for a 2-D case:

$$\boldsymbol{U}_{i,j}^{n+1} = \boldsymbol{U}_{i,j}^{n} - \frac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+1/2,j} - \boldsymbol{F}_{i-1/2,j}\right) - \frac{\Delta t}{\Delta y}\left(\boldsymbol{G}_{i,j+1/2} - \boldsymbol{G}_{i,j-1/2}\right), \tag{6.1}$$

We would like to be able to implement easily any conservative scheme in the Eq. (6.1) form, at least for first- and second-order methods in space and time. We thus want to easily change the numerical fluxes definition, i.e., to easily change the cell-average-based reconstruction method, see Sec. 3.2.2. The scheme implementation must then be able to treat any number of neighbors, the numerical method for pressureless gas possibly leading to a numerical flux depending on all the grid neighbors of the current cell, see Sec. 3.2.2. For example in 2-D we have, for the Bouchut flux:

$$\begin{aligned} F_{i+1/2,j}^{(1)} \quad = \quad F(\quad & m_{i,j}^{n}, m_{i-1,j}^{n}, m_{i-1,j-1}^{n}, m_{i-1,j+1}^{n}, m_{i,j-1}^{n}, m_{i,j+1}^{n} \\ & u_{i,j}^{n}, u_{i-1,j}^{n}, u_{i-1,j-1}^{n}, u_{i-1,j+1}^{n}, u_{i,j-1}^{n}, u_{i,j+1}^{n}). \end{aligned} \tag{6.2}$$

Furthermore, in order to obtain the second order in time, the scheme could contain several steps. For example using a Runge-Kutta technique, the scheme would become:

$$\begin{aligned} \boldsymbol{U}_{i,j}^{\star} &= \boldsymbol{U}_{i,j}^{n} - \tfrac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+1/2,j} - \boldsymbol{F}_{i-1/2,j}\right) - \tfrac{\Delta t}{\Delta y}\left(\boldsymbol{G}_{i,j+1/2} - \boldsymbol{G}_{i,j-1/2}\right), \\ \boldsymbol{U}_{i,j}^{\star\star} &= \boldsymbol{U}_{i,j}^{\star} - \tfrac{\Delta t}{\Delta x}\left(\boldsymbol{F}_{i+1/2,j}^{\star} - \boldsymbol{F}_{i-1/2,j}^{\star}\right) - \tfrac{\Delta t}{\Delta y}\left(\boldsymbol{G}_{i,j+1/2}^{\star} - \boldsymbol{G}_{i,j-1/2}^{\star}\right), \\ \boldsymbol{U}_{i,j}^{n+1} &= \frac{\boldsymbol{U}_{i,j}^{\star} + \boldsymbol{U}_{i,j}^{\star\star}}{2} \end{aligned} \tag{6.3}$$

Such an evolution of the computed fields must be possible. Moreover, we want to use the scheme in various configurations from 1-D to fully 3-D schemes and 3-D with dimensional splitting, see Sec. 3.2.2. This also require a great genericity in the numerical scheme implementation.

Transport algorithm

Our scheme implementation must then satisfies the two main constraints previously described, i.e., to allow extension to any conservative scheme given by Eq. (6.1), and to allow any time evolution based on explicit Euler steps. In order to make the method evolve easily, we separate the different steps of the finite volume scheme:

- cell reconstruction,
- flux computation,
- field update.

leading to the following algorithm:

Algorithm 9 Transport timestep procedure

```
procedure Transport_timestep(Spray)
...
{Reconstruction step}
call Reconstruction(Recons, Slope, Mean, Spray)
...
{Fluxes computation}
call Flux_evaluation( Flux,Recons,Slope,Mean,Spray)
...
{Fields evolution}
call Update_field(Spray, Flux)
...
end procedure Transport_timestep
```

We analyze here, for these different contributions, the way used to view the most generic method for a generic computational domain.

Cell reconstruction The cell reconstruction step lead to the computation of the reconstructed values needed for the flux computation, see Eq. (3.34) for example. In a first order scheme, it simply corresponds to the cell average, the reconstruction being piecewise constant. For a second order scheme, we need to first compute a slope, the reconstruction being piecewise linear, see for example Eq. (3.53). The reconstruction procedure is then divided into two sub-procedures devoted to slope (Slope), and reconstruction (Recons) computations. We also compute, together with the slope, the bar values (Mean), given in Eq. (3.49):

Algorithm 10 Reconstruction procedure

```
procedure Reconstruction(Recons, Slope, Mean, Spray)
...
{Slope computation}
call Slope_comp(Slope, Mean, Spray)
...
{Reconstruction computation}
call Recons_comp(Recons,Slope,Mean)
...
end procedure Reconstruction
```

Up to this level, the structure of the data is transparent, and any modification on the arrays contained in the different data structures do not induce any modification in the transport scheme. The different arrays of the structures are used inside the procedures Slope_comp and Recons_comp where we define specific array-based procedures, for example for the slope computation:

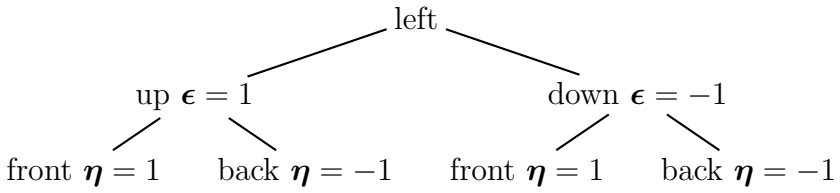
Algorithm 11 Slope procedure

```
procedure Slope_comp(Slope, Mean, Spray)
...
{Size moments}
call Slope_sizes(Slope%sizes, Mean%sizes, Spray%sizes)
...
{Velocity moments}
call Slope_vel(Slope%vel, Mean%vel, Spray%vel)
...
{Crossed Size-Velocity moments}
call Slope_sizes_vel(Slope%sizes_vel, Mean%sizes_vel, Spray%sizes_vel)
...
{Thermodynamical temperature moments}
call Slope_temptherm(Slope%temptherm, Mean%temptherm, Spray%temptherm)
...
end procedure Slope_comp
```

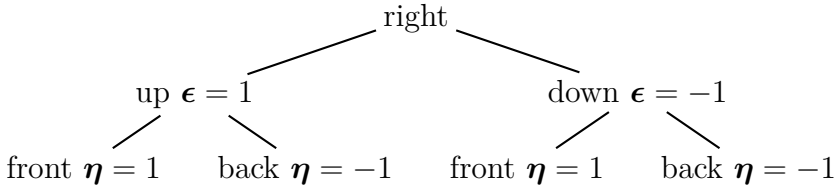
We can thus easily add new fields in the structures and make the spray model along with the numerical method evolve. Besides, several types of slope limiters are available and easily used in the different slopes procedures.

We now focus on the computational domain evolution, in order to have generic, slope and reconstruction, procedures for any type of domain, from 1-D to 3-D configurations. Regarding the slope computation, the genericity for the computational domain is contained in the data structure, Slope%array$(N_x, N_y, N_z, N_S, N_{\text{mom}}, N_{\text{dim}})$, where we can store the number of components needed for the considered dimension. Indeed, we store slopes for x, y and z dimensions, in a 3-D case, in the last column of the slope array. The slope computation in itself does not change with the dimension of the computational domain, the slope computed being always one-dimensional, therefore no extra work is needed to handle multi-dimensional cases. On the other hand, the reconstruction computation evolves with the number of dimensions of the domain. Indeed, as we can see on Fig. 6.2, the form and the number of the reconstruction evolve with the number of dimensions. In order to implement these reconstruction in a general form for 1-D to 3-D configurations, we propose Algo. 12, where the complexity lies in the function linear_part, and where dx(:) $= (\Delta x, \Delta y, \Delta z)$, and slope(:) $= (slope_x, slope_y, slope_z)$.

Flux computation The same features are to be found for the flux computation procedure. Indeed we have a generic flux, as far as computational domain dimension is concerned, thanks to the data structure: Flux%array $(N_x, N_y, N_z, N_S, N_{\text{mom}}, N_{\text{dim}})$, so that we store in the array the spatial components of the flux. We also define procedures for the different fields of the structure in Algo. 13.



$$\text{recons}^{2i-1} = field_d^{2i-1} - \frac{\Delta x}{2}\,\text{slope}_x^{2i-1} + \boldsymbol{\epsilon}\,\frac{\Delta y}{2}\,\text{slope}_y^{2i-1} + \boldsymbol{\eta}\,\frac{\Delta z}{2}\,\text{slope}_z^{2i-1} \tag{6.4}$$



$$\text{recons}^{2i-1} = field_d^{2i-1} + \frac{\Delta x}{2}\,\text{slope}_x^{2i-1} + \boldsymbol{\epsilon}\,\frac{\Delta y}{2}\,\text{slope}_y^{2i-1} + \boldsymbol{\eta}\,\frac{\Delta z}{2}\,\text{slope}_z^{2i-1} \tag{6.5}$$

Figure 6.2 – *Formulation for second order transport scheme reconstruction in any space dimension.*

Algorithm 12 Multi-dimensional reconstruction

```
procedure Recons_comp(Recons,Slope,Mean)
...
for i=1 to 2N_dim do
  Recons(i) := Mean(i) + linear_part(i,Slope(:))
end for
end procedure Recons_comp
...
function linear_part(n,slope)
{1-D case}
if (n odd) then
  linear_part := dxi(1)/2*slope(1)
else if (n even) then
  linear_part := -dxi(1)/2*slope(1)
end if
{2-D to 3-D case}
p := 2
while (p≤ N_dim) do
  if (n=1 or n=2) or ((n=5 or n=6) and p=2) or ((n=7 or n=8) and p=3) then
    linear_part := linear_part+dx(p)/2*slope(p)
  else if (n=3 or n=4) or ((n=7 or n=8) and p=2) or ((n=5 or n=6) and p=3) then
    linear_part := linear_part-dx(p)/2*slope(p)
  end if
    p := p+1
end while
end function linear_part
```

Algorithm 13 Flux computation

procedure Flux_comp(Flux, Slope, Recons)
...
{Size moments}
call Flux_sizes(Flux%sizes, Slope%sizes, Recons%sizes)
...
{Velocity moments}
call Flux_vel(Flux%vel, Slope%vel, Recons%vel)
...
{Crossed Size-Velocity moments}
call Flux_sizes_vel(Flux%sizes_vel, Slope%sizes_vel, Recons%sizes_vel)
...
{Thermodynamical temperature moments}
call Flux_temptherm(Flux%temptherm, Slope%temptherm, Recons%temptherm)
...
end procedure Flux_comp

Besides, we have, as low level procedures, various types of fluxes available: cartesian, axisymmetric, first order, second order, etc.

Field evolution Finally, when flux computation is over, we need to make the computational fields evolve, see Algo. 9. We use the procedure Update_field to compute the fields at time t^{n+1}, with the formula Eq. (6.3). In order to define a procedure generic for 1-D to 3-D domains, we set the procedure euler, called inside a loop over the direction of the domain done in the procedure Update_field, in Algo. 14.

Algorithm 14 Field time evolution

procedure Update_field($spray^n$,$spray^{n+1}$)
...
$spray^\star = spray^n$
for i=1 to N_{dim} **do**
  call euler($spray^\star$,i)
end for
$spray^{n+1} = spray^\star$
...
end procedure field_update
...
procedure euler($spray^\star$,m)
{Evolution for space direction m}
$spray^\star = spray^\star - \Delta t/dx(m) * (Flux_{i+1/2}(m) - Flux_i(m))$
end procedure euler

It can then be used whatever the number of directions of the computational domain is. This euler procedure is used for all the spray arrays, `sizevel, momvel, temptherm`... For second order evolution, it may be necessary to use a Runge kutta scheme, see Eq. (6.3). In

that case, the runge kutta algorithm is handled at the beginning of the physical transport scheme. We implement it in a way easily extendable to other two steps, or more, methods. The physical transport timestep implementation is given in Fort. 10. One can notice the use of operator overloading, defined in the previous paragraph, allowing to manipulate directly the spray structure without needing to write its different components. We use here the overloading of the affectation `(=)`, the addition `(+)` and the division `(/)` operators, see for example:

```
Spray = (Sprayarray(1_ip)+Sprayarray(Nstep+1_ip))/real(Nstep,kind=rp)
```

Dimensional splitting

As we explained in Sec. 3.2.2, we will mainly use dimensional splitting for the transport in physical space. This dimensional splitting, as the operator splitting, can take several forms, from Lie to the different Strang splittings. Once again, we look for a performant generic implementation. We make use of the algorithm already defined for operator splitting, to write:

Algorithm 15 Dimensional splitting

for i = 1 to N_{step} **do**
  call phys_transp(spray,step_dim(i),step_dt(i))
end for

The variable N_{step} represents the number of step of the splitting algorithm while step_dim(i) is the direction of transport and step_dt(i) the timestep of the current step. We can then choose, with this algorithm, any ordering for the transport steps, and we can alternate the ordering from one step to another, as required by the "modified" splitting algorithm, see Sec. 3.2.2. Three elements need then to be first determined: the scalar N_{step}, the vectors step_dim and step_dt. For the Lie and the "modified" Strang splitting, $N_{\text{step}} = N_{\text{dim}}$. For the "classical" Strang algorithm, $N_{\text{step}} = 2\ N_{\text{dim}} - 1$, refer to Sec. 3.2.2 for details. For the Lie and "modified" Strang splitting, we can easily create step_dim, using the Fortran vector constructor together with implicit loop:

```
step_dim(:)  = (/ (i,i=1_ip,grid%Ndim) /)
```

With this step_dim definition, the steps are performed first in x direction, then in y direction and finally z direction, if 3-D. We can modify the order of these steps, shifting the step_dim array with the Fortran intrinsic function `cshift`, realizing circular permutation:

```
step_dim(:)  = cshift((/ (i,i=1_ip,grid%Ndim) /),shift=shift_val)
```

This shifting operation will be needed at the beginning of each iteration for the "modified" Strang splitting algorithm, where we alternate the order of the steps. It appears a bit more complicated for the "classical" Strang algorithm, where step_dim takes the general form step_dim=(a,b,c,b,a), with a, b and c $\in \{1, 2, 3\}$. In order to determine the form of the algorithm, the first (a) and the center (c) directions are given as parameters. The array is then constructed from this information: we first create a vector, excluding the center direction of the algorithm from the vector N=(1 ... N_{dim}), thanks to the Fortran array extraction function `pack`:

Fortran 10 Physical transport subroutine

```
subroutine Physic_transp_dt(Spray)
type(field_5d), intent(inout)  ::  Spray
!-local variables
type(field_5d), dimension(:), allocatable  ::  Sprayarray
integer(kind=ip) ::  i,Nstep
!-Choice of time scheme Euler or RK2
if (param%type_timescheme ==  1_ip) then !Euler
  Nstep = 1_ip
elseif (param%type_timescheme==  2_ip ) then !RK2
  Nstep = 2_ip
endif
!
!-TIMESTEP CHOICE-----------------------------------------------------|
if (Nstep==1_ip) then
  !-EULER TIMESTEP----------------------------------------------------|
  call physic_transp_euler(Spray)
  !-------------------------------------------------------------------|
  !
elseif (Nstep==2_ip) then !RK2 (generic routine for multi-step time method)
  !-RK2 TIMESTEP------------------------------------------------------|
  !-First call--------------------------------------------------------|
  call physic_transp_euler(Spray)
  !-------------------------------------------------------------------|
  !-Allocations-------------------------------------------------------|
  !-Array allocation & array component allocation---------------------|
  allocate(Sprayarray(Nstep+1_ip))
  call allocate(Sprayarray,0_ip)
  !-Loop on the step--------------------------------------------------|
  Sprayarray(1_ip)  = Spray
  do i=1_ip,Nstep
    Sprayarray(i+1_ip)  = Sprayarray(i)
    !-Second (and more if order>2)--------------------------------|
    call physic_transp_euler(Sprayarray(i+1_ip))
    !-------------------------------------------------------------|
  enddo
  !-Final Form--------------------------------------------------------|
  Spray = (Sprayarray(1_ip)+Sprayarray(Nstep+1_ip))/real(Nstep,kind=rp)
  !-------------------------------------------------------------------|
  !-array component deallocation & array deallocation-----------------|
  call deallocate(Sprayarray)
  deallocate(Sprayarray)
  !
endif
!
end subroutine Physic_transp_dt
```

```
step_dimtemp(:) = pack((/(i,i=1_ip,Ndim)/),&
                  mask=(/(i/=param%split_dim_ctr,i=1_ip,Ndim)/))
```

This extraction gives step_dim=(a,b), if a<b or step_dim=(b,a), if a>b. In the second case, we apply a circular shift, through `cshift`, to the vector `step_dimtemp` to fit the first direction asked. We can then completely fill the vector step_dim, the vector step_dimtemp giving the first and last part of the splitting algorithm.

```
step_dim(1_ip:Ndim-1_ip)  = step_dimtemp(:)
step_dim(Ndim+1_ip:Nstep) = step_dimtemp(Ndim-1_ip:1_ip:-1_ip)
step_dim(Ndim)            = param%split_dim_ctr
```

where `param%split_dim_ctr`, is the center direction of the algorithm. This procedure seems quite heavy, nevertheless it never presumes the value of N_{dim}. On the other hand, the structure of the array step_dt is rather simple, and described in Algo. 16. The idea is then

Algorithm 16 Splitting timestep array

{Lie and Strang "modified" algorithm}
step_dt(:) = Δt
{Strang "classical" algorithm}
if (i /= N_{dim}) **then**
  step_dt(i) := $\Delta t/2$
else
  step_dt(i) := Δt
end if

to be able to use the 1-D scheme implementation for the dimensional splitting, where we have 3-D arrays and where the transport direction is not known *a priori*. We introduce the variable N_{dim_transp}, taken equal to one for dimensional splitting case and to N_{dim} otherwise. The flux expressions are computed with N_{dim_transp} instead of N_{dim} to have a 1-D flux for dimensional splitting. Moreover we define td as being the number of the transport direction, given by the argument step_dim(i). We then perform the one-dimensional transport in this direction. Finally, to extend the 1-D flux computation procedure to the dimensional splitting case, we need to compute the fluxes for the velocity components different from the transport component. To do so we set the array od with N_{dim}-1 elements and the 1-D flux procedure for velocity moments, mass (zeroth order moment) and momentum (first order moment), becomes Algo. 17. This procedure can be used for 1-D as well as for 3-D dimensionally split configurations.

6.3.3 Radau5 Program modular introduction

We choose to use, for the integration of the ODE system coming from phase space transport, the Radau5 program written by Hairer, (Hairer and Wanner 1996), providing an implementation of Radau IIA ODE solver, detailed in Sec. 3.2.3. This program was written in Fortran77, and contains three elements, a driver preparing the call to the Radau5 subroutine containing the solver. The third element is an external subroutine used to define ψ function, the program solving the equation:

$$\mathrm{d}_t \, \boldsymbol{Y} = \psi(t, \boldsymbol{Y}). \tag{6.6}$$

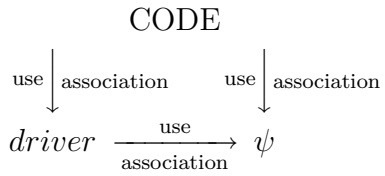
Algorithm 17 Combined 1-D and dimensional splitting formulation

```
procedure Flux_vel_1D(fluxvel,slopevel,reconsvel,td)
{Creating a vector with the directions without the splitting direction}
tabledim = (1,...,N_dim)
od := tabledim-td {excluding transport direction td from tabledim}
{Computing mass flux}
call mass_flux_1D(flux_mass,recons_mass,slope_mass)
{Computing velocity flux}
{Splitting difficulty : 2 different expressions}
{=> one for the flux of the transport direction}
{=> one for the other(s) (2D or 3D) direction(s)}
{Computing transport direction flux}
call vel_flux_1D(flux_vel(td),recons_vel(td),slope_vel(td))
{Computing other(s) direction(s) (if N_dim>1)}
{standard scalar moment transported by spray}
if (ndim>1) then
  for i=1,N_dim-1 do
    call scal_mom_flux_1D(flux_vel(od(i)),recons_vel(od(i)),slope_vel(od(i)))
  end for
end if
end procedure Flux_vel_1D
```

It was not decided to realize a new implementation in Fortran90/95 of this solver given the important work it represents, for development and validations. In order to introduce this program in our computational code, we decide to rewrite the driver and to propose a new way to implement the function ψ. On the other hand, we do not want to modify the solver itself, since it is a very efficient implementation. The driver is written as a procedure called by the code to launch the Radau solver. Moreover, we create a module `drop_model`, to rewrite in Fortran90/95 the function ψ implementation. The first impact of Fortran 90 implementation is the transfer of data between the driver and the function ψ. Indeed, all the physic of the model is contained in the function ψ. It then needs to access to many variables of the code, and this can be done by module use association between its host module, `drop_model` and the code modules containing desired information. Furthermore, the function ψ may need to access to local variables defined in driver or more generally in its host module, that contains the phase space solver, `Transp_phase`. For example, the phase evolution being local in space, we need to know, in ψ, the local gas properties for each call of the solver. This transfer can now be easily done through a module association between the driver module and the function module. We can, for example define variables in `drop_model` to store local gas velocity, and fill them in the module `Transp_phase`. Consequently all the transfers between the solver driver and the right-hand side function ψ are done through module association, as explained in Fig. 6.3. In a Fortran77 framework, all these transfers are done using the Radau solver, the data being transfered from the driver to Radau and from Radau to the function by input arguments, see (Hairer and Wanner 1996). This prevent to have a generic driver, the driver being written for one precise function. The driver procedure becomes then a Fortran90/95 interface for the Radau solver. It realizes a data encapsulation defining and dynamically allocating all the local variables needed for

$$driver \xrightarrow[\text{arguments}]{\text{procedure}} radau\_solver \xrightarrow[\text{arguments}]{\text{procedure}} \psi$$

(a) Classical information transfer Radau F77



(b) New information transfer through module association

Figure 6.3 – *Scheme explaining data exchanges between the Radau driver and function ψ. (a) the arguments are passed only through the Radau procedure; (b) the data transfer uses only F90 module association, ψ receiving data from driver for local data, and from the rest of the CODE for global data.*

Radau, and taking as arguments all the Radau variables needing to be inherited. It allows then a generic call for Radau:

```
call driver_radau(y,t0,tend,nt,tol)
```

allowing to drive the main input/output arguments of the solver: the variable vector `y`, the beginning and ending time `t0`,`tend` along with the number of points in this interval, `nt`. The argument `tol` is an instance of the derived type `type_tol` defined by:

```
type tol_type
real(kind=rp), dimension(:), allocatable ::  rtol, atol
end type tol_type
```

containing the arrays for the absolute and relative tolerances needed by the Radau procedure. This procedure allows an easy and safe use of the Radau solver. Regarding the ψ function implementation, we take advantage of the new module, `drop_model`, definition to implement a new generic procedure allowing to choose between several right-hand sides. Indeed the Radau procedure call the procedure `fed`, that we define in Fort. 11. In that case, the argument `ipar`, generally used to pass variables to the model from the driver procedure, is used as a flag determining spray modeling to be used. It allows to define in the same modules all the types of model we wish to use. One can note that implicit profiles can not be used for the arguments `y` and `f` because this subroutine is called from a Fortran77 procedure, enforcing the explicit declaration of the dimension: `dimension(n)`.

We thus decouple in this approach the solver and the physical models involved, defining a generic driver for Radau and interfacing the solver with a generic procedure connecting various models. This modular implementation allows easy extension of modeling as well as easy sharing of the solver for other applications.

Fortran 11 Radau external subroutine

```
subroutine fed(n,t,y,f,rpar,ipar)
!-Right-hand side of multi-fluid equation----------------------------!
integer(kind=ip), intent(in) ::  n
real(kind=rp), dimension(n), intent(in)  ::  y
real(kind=rp), dimension(n), intent(out) ::  f
real(kind=rp), dimension(1_ip), intent(in)  ::  rpar
integer(kind=ip), dimension(2_ip), intent(in) ::  ipar
real(kind=rp), intent(in)     ::  t
!-Choose type of equation solved-------------------------------------!
f=0._rp
!-Multi-size moments model-------------------------------------------!
if (ipar(1_ip)==1_ip) then
  call mms_eq(t,y,f)
!-Multi-Fluid model--------------------------------------------------!
elseif (ipar(1_ip)==2_ip) then
  call mf_eq(t,y,f)
!-Collisional Multi-Fluid model--------------------------------------!
elseif (ipar(1_ip)==3_ip) then
  call mv_col(t,y,f)
endif
end subroutine fed
```

6.4 Separated Boundary condition implementation

In order to ensure an easy increase of numerical schemes implemented and of type of configurations simulated, we must isolate the treatment of boundary conditions from the scheme implementation. We want to store the ghost cells for each frontier of the domain, in the composed class `comp_field_4d` defined in Sec. 6.2. We explain here the way chosen to handle the boundary conditions and to couple them with the physical transport implementation.

6.4.1 Obtaining generic boundary conditions

We focus here on one frontier, studying the way to fill a boundary condition type, `field_4d`, containing the ghost cells at the considered frontier. We deal here with three types of basic boundary conditions. In an injection case, the content of the ghost cells at the boundary is read in a previously written file, that must be created in the initialization of the run or of the iteration, depending whether the injection conditions are stationary or not. On the other hand, for periodicity and symmetry conditions, the content of the ghost cells must be extracted from the concerned field. We then define a generic procedure `extract_field`, called as follows:

call extract_field(Field, Field_extract, bound_dim, bound_shape, pos)

In the case of a five-dimension array, coming from the spray type variable, `field_5d`, this procedure extracts, from the Field($N_x, N_y, N_z, N_S, N_{\text{mom}}$), the four-dimension array Field_extract($N1, N2, N_S, N_{\text{mom}}$). This extraction is done by taking only the point pos

for the dimension bound_dim. The resulting space dimension: $(N1, N2)$ are given in the procedure argument bound_shape=$(N1, N2)$. In other word, if we take for example bound_dim equal to one in the case of an initial five-dimension array:

$$\text{Field\_extract}(:,:,:,:) = \text{Field}(\mathbf{pos}, :,:,:,:)$$

This procedure is implemented generically through the use of the Fortran intrinsic `pack` function, and can deal with bound_dim from one to three. The idea of this function is to generate a logical mask where the elements we wish to drop are replaced by false elements. We create this mask with an implicit loop combined with a logical condition on the loop index `i/=pos`. It gives the subroutine written in Fort. 12. We define similar subroutines for five- and four-dimension arrays, and we regroup them under a generic subroutine `extract_field`. These routines can treat any of the model moments, `sizes`, `vel`, `sizes_vel` and `temptherm`. We obtain with this extracted field the ghost cells at the considered frontier i:

$$(Ghostcell)_{\text{frontier } i} = Field\_extract \tag{6.7}$$

This extraction is sufficient to obtain the ghost cells for periodic boundary conditions, where we extract the opposite side in the considered dimension. For example if we want the ghost cells at $x = -\Delta x$ for a periodic boundary condition, we take the corresponding field at Nx:

$$(\text{Ghostcell\_periodic})_{x=-\Delta x} = \text{Field}(\mathbf{Nx}, \text{1:Ny}, \text{1:Nz}, \text{1:Ns}, \text{1:Nmom})$$

On the other hand, to obtain the ghost cells for the symmetry boundary condition, we take the immediate neighbor, i.e., for $x = -\Delta x$, the first point in x direction:

$$(\text{Ghostcell\_symmetry})_{x=-\Delta x} = \text{factor*Field}(\mathbf{1}, \text{1:Ny}, \text{1:Nz}, \text{1:Ns}, \text{1:Nmom})$$

Furthermore, we need to add the multiplicative array factor. Actually, although the extraction is sufficient to symmetrize scalar fields as mass, a second operation is to be completed to generate symmetry boundary condition for vectorial field, as velocity. Indeed, we need fill the ghost cells with the opposite value for the component normal to the symmetry vector or plane. We then multiply the extracted array by factor, an array with the shape of the boundary and whose elements take the values +1 or -1. The multiplication is done element by element. We then want to construct this factor array, for an arbitrary number of velocity moments and for an arbitrary number of space directions. In order to illustrate this array construction, we take an example for two space directions and four velocity moments, order zero to three. At a fixed point of the physical space and for a fixed size, we have the array of moments A of Eq. (6.8),

$$A = \begin{pmatrix} m^{(0)} \\ m_x^{(1)} \\ m_y^{(1)} \\ m_x^{(2)} \\ m_y^{(2)} \\ m_x^{(3)} \\ m_y^{(3)} \end{pmatrix} \xrightarrow[x=0]{Symmetry} A_{symm} = \begin{pmatrix} m^{(0)} \\ -m_x^{(1)} \\ m_y^{(1)} \\ -m_x^{(2)} \\ m_y^{(2)} \\ -m_x^{(3)} \\ m_y^{(3)} \end{pmatrix} \Rightarrow factor = \begin{pmatrix} 1 \\ -1 \\ 1 \\ -1 \\ 1 \\ -1 \\ 1 \end{pmatrix} \tag{6.8}$$

where the first moment $m^{(0)}$ is a scalar, the mass, and the moments $m^{(1)}$ to $m^{(3)}$, velocity moments of order one to three, are vectors with N_{dim} components. In the example of a symmetry condition at $x = 0$, we put in the ghost cells, the array A_{symm} of Eq. (6.8), taking

Fortran 12 Boundary extraction routine

```
subroutine extract_field5d(Field,Field_extract,ndim,N,pos)
!-subroutine to extract from Field(Nmom,Nsec,Nx,Ny,Nz) excluding one space
!-dimension into the array :
!-Field_extract(Nmom,Nsec,N1,N2).
!-Argument ndim is the dimension left
real(kind=rp), dimension(:,:,:,:,:), intent(in) :: Field
real(kind=rp), dimension(:,:,:,:),   intent(inout):: Field_extract
integer(kind=ip),dimension(10_ip), intent(in) ::  DebFin
integer(kind=ip), dimension(4_ip),   intent(in):: N
integer(kind=ip), intent(in)  :: ndim, pos
!-local variables
logical, allocatable, dimension(:,:,:,:,:)  ::  mask
real(kind=rp), allocatable, dimension(:) ::  extracted
integer(kind=ip)   ::  i, j, k, m, sec, Nmom, Nsec, Nx, Ny, Nz, N1, N2
!-affectations
Nmom = size(Field,1_ip); Nsec = size(Field,2_ip);Nx = size(Field,3_ip)
Ny = size(Field,4_ip) ; Nz = size(Field,5_ip); N1 = N(1_ip); N2 = N(2_ip)
!-allocation mask has the size of Field
allocate(mask(1:Nmom,1:Nsec,1:Nx,1:Ny,1:Nz))
! extracted is the vector resulting : dim(Nmom*Nsec*N1*N2)
allocate(extracted(Nmom*Nsec*N1*N2))
!-Defining mask: logical=True if value extracted, False otherwise
if (ndim == 1_ip) then
  mask  = reshape( source  =(/  (((((  i==pos,  m   = 1,Nmom),    &
                                       sec = 1,Nsec),    &
                                       i   = 1,Nx),    &
                                       j   = 1,Ny),    &
                                       k   = 1,Nz) /), &
                   shape   =(/ Nmom, Nsec, Nx, Ny, Nz /))
elseif (ndim  ==2_ip) then
  mask  = reshape(  source =(/  (((((  j==pos,  m   = 1,Nmom),    &
                                       sec = 1,Nsec),    &
                                       i   = 1,Nx),    &
                                       j   = 1,Ny),    &
                                       k   = 1,Nz) /), &
                    shape  =(/ Nmom, Nsec, Nx, Ny, Nz /))
elseif (ndim ==3_ip) then
  mask  = reshape(  source =(/  (((((  k==pos,  m   = 1,Nmom),    &
                                       sec = 1,Nsec),    &
                                       i   = 1,Nx),    &
                                       j   = 1,Ny),    &
                                       k   = 1,Nz) /), &
                    shape  =(/ Nmom, Nsec, Nx, Ny, Nz /))
endif
!-Extracting
extracted     = pack(Field,mask)
Field_extract = reshape(  extracted,  (/ Nmom, Nsec, N1, N2 /)  )
!-deallocate
deallocate(mask,extracted)
end subroutine extract_field5d
```

the opposite of all the x-components. The factor array takes the value -1 for the components of the vectorial fields in the direction normal to the symmetry vector or plane. The problem is to know the number of the components with knowing only their index in the array A for any number of moments or of directions. For the mass, we have only factor(1)=1. Let us restrict to the velocity moments and write their array index, the number of the moment and the number of the component for the current element:

$$\begin{array}{cccc} \text{Moments} & \text{Array index} & \text{Moment number} & \text{Component number} \\ \begin{pmatrix} m_x^{(1)} \\ m_y^{(1)} \\ m_x^{(2)} \\ m_y^{(2)} \\ m_x^{(3)} \\ m_y^{(3)} \end{pmatrix} & \begin{pmatrix} 2 \\ 3 \\ 4 \\ 5 \\ 6 \\ 7 \end{pmatrix} & \begin{pmatrix} 2 \\ 2 \\ 3 \\ 3 \\ 4 \\ 4 \end{pmatrix} & \begin{pmatrix} 1 \\ 2 \\ 1 \\ 2 \\ 1 \\ 2 \end{pmatrix} \end{array} \tag{6.9}$$

We want to obtain for a current index $p > 1$, the number of the component n_{comp}, of the current moment n_{mom}, for an arbitrary N_{mom} number of moments and N_{dim} number of dimensions of the domain. One can verify on the previous example the following relation:

$$p = (n_{\text{mom}} - 2)N_{\text{dim}} + n_{\text{comp}} + 1, \tag{6.10}$$

giving

$$n_{\text{comp}} = p - 1 - (n_{\text{mom}} - 2)N_{\text{dim}}. \tag{6.11}$$

We thus need, to know the number of the component, the number of the moment n_{mom} for the current index p:

$$n_{\text{mom}} = ceiling\left(\frac{p-1}{N_{\text{dim}}}\right) + 1, \tag{6.12}$$

where $ceiling(x)$, with x real, takes the smaller integer greater than x. Finally we have:

$$n_{\text{comp}} = p - 1 - (ceiling\left(\frac{p-1}{N_{\text{dim}}}\right) - 1)N_{\text{dim}}. \tag{6.13}$$

This relation gives the number of the component of velocity moments as a function of the array index and thus allows us to construct the array factor. We decide again to use the `pack` Fortran intrinsic function: we create a logical mask where false elements will take the value -1 and the true elements +1. The logical condition necessary for the mask is then:

(p == -1) or (n_comp /= normal_sym_dir)

where normal_sym_dir is the index if the direction normal to the symmetry vector or plane. Its Fortran implementation is given in Fort. 13. where we create the logical mask `mask_factor_vel` with an implicit loop with the appropriate logical test and a reshape function to have the good array shape. Then we apply this mask to a conformable array identically equal to one, and we replace the false elements of the mask by -1. We created here a procedure to construct the array factor allowing to construct the ghost cells from the velocity moments of the spray, `velmom` variables at the frontier. This procedure works for any number of moments and space directions. It is only restricted to vector of moments with the structure of A in Eq. (6.8), with one scalar moment before the vectorial moments.

Fortran 13 Extraction for symmetry boundary

```
!complex case : -1 for vel in direction of BC for vel and sizes_vel
!creating an array factor_vel(Nmom,Nsec,N1,N2) with 1 for 0th order
!velocity moment and axi component (ie componet in the axis or plane of
!symmetry) of the higher order velocity moment; and -1 for component
!perpendicular to the axis (plane) of symmetry of moment of order >=1.
!
!-Velocity moments-----------------------------------|
Nmom  = grid%Nmom_vel
Nsec  = grid%Nsec
!-remind N = bound_shape
N1    = N(1_ip)
N2    = N(2_ip)
allocate(factor_vel(Nmom,Ns,N1,N2))
allocate(mask_factor_vel(Nom,Ns,N1,N2))
mask_factor_vel = reshape ( source = (/ (((( p==1_ip .or.                  &
                           p-1_ip-((ceiling(real(p-1_ip,kind=rp)           &
                           /real(Ndim,kind=rp))-1_ip)*Ndim)/=nbound(1_ip), &
                           p   = 1_ip,Nmom),    &
                           sec = 1_ip,Nsec),    &
                           i   = 1_ip,N1),    &
                           j   = 1_ip,N2) /), &
                           shape   =(/ Nmom,Ns,N1,N2 /)  &
                          )
factor_vel  = unpack( (/ (1_ip,i=1_ip,Nmom*Ns*N1*N2) /), &
                     mask=mask_factor_vel,field=-1_ip)
deallocate(mask_factor_vel)
```

If we want to deal with another form of moment vector, we just need to change the logical condition, defining the relation Eq. (6.13) between the index of the vector and the number of components of vectorial fields. This procedure is defined for spray variables and have been extended, with the same ideas, to treat variables of the reconstruction process, `Slope` and `Recons`.

6.4.2 Coupling transport scheme with boundary conditions

These ghost cells generation is needed within the transport scheme. Indeed between each step of the transport scheme, we need to generate the ghost cells needed for the next step. In this way, we need to create ghost cells for the spray variables to compute the slopes, the slope of the cell i needing the values of the neighbor cells, as well as the flux computation needs ghost cells for mean, slope and reconstruction variables and finally the field update needs ghost cells for the flux variables. We then complete the transport algorithm with these ghost cells computation, see Algo. 18. The procedure Step_bound, Algo. 19,

Algorithm 18 Transport Algorithm

procedure Transport_timestep(Spray)
...
{Spray ghost cells for reconstruction procedure}
call Recons_bound(Bound_spray,Spray)
{Reconstruction step}
call Reconstruction(Recons, Slope, Mean, Spray)
...
{Mean, Slope and Recons ghost cells for flux computation procedure}
call Flux_bound(Bound_mean,Bound_slope,Bound_recons,Mean, Slope, Recons)
{Fluxes computation}
call Flux_evaluation(Flux,Recons,Slope,Mean,Spray)
...
{Flux ghost cells for flux update field procedure}
call Update_bound(Bound_flux, Bound_mean, Bound_slope, Bound_recons, Bound_spray)
{Fields evolution}
call Update_field(Spray, Flux)
...
end procedure Transport_timestep

creates the variable Bound_Var, containing the ghost cells of all the computational domain frontiers. Ghost cells of each frontier, extracted from the field Var, as previously explained, are stored in the array Bound_Var%c, Bound_Var being of type `comp_field4d`, i.e., each of the component of the array `c` is of type `field_4d`, see Sec. 6.2. The inlet and outlet condition of the frontier i for the generic space w-direction, representing x- y- or z-direction, are then stored the following way:

$$\begin{cases} Bound\_Var\%c(2i-1) &= (Ghostcell)_{w_i=-\Delta w_i} \\ Bound\_Var\%c(2i) &= (Ghostcell)_{w_i=w_{\max i}+\Delta w_i} \end{cases} \tag{6.14}$$

This procedure generates the ghost cells for all the domain frontiers, whatever the number of spatial directions is. We have therefore Algo. 19. The vector type_bound($2N_{\text{dim}}$)

Algorithm 19 Bound_var creation

```
procedure Step_bound(Bound_Var,Var,type_bound)
...
for i=1 to N_dim do
  {Generate inlet condition}
  call fill_bound_var(Bound_Var(2i-1), Var, 2i-1)
  {Generate outlet condition}
  call fill_bound_var(Bound_Var(2i), Var, 2i)
end for
...
end procedure Step_bound
```

defines the type of the boundary condition for each frontier (periodic, symmetry, etc). We pass, as the last input argument of fill_bound_var, the number of the current frontier. We then have, in the example of a 3-D domain $[0, X_{\max}] \times [0, Y_{\max}] \times [0, Z_{\max}]$:

$$\left\{ \begin{array}{lcl} Bound\_Var\%c(1) & = & (Ghostcell)_{x=-\Delta x}. \\ Bound\_Var\%c(2) & = & (Ghostcell)_{x=X_{\max}+\Delta x}. \\ Bound\_Var\%c(3) & = & (Ghostcell)_{y=-\Delta y}. \\ Bound\_Var\%c(4) & = & (Ghostcell)_{y=Y_{\max}+\Delta y}. \\ Bound\_Var\%c(5) & = & (Ghostcell)_{z=-\Delta z}. \\ Bound\_Var\%c(6) & = & (Ghostcell)_{z=Z_{\max}+\Delta z}. \end{array} \right. \tag{6.15}$$

To illustrate this algorithm we give in Fort. 14 the Fortran subroutine `Recons_bound` generating ghost cells needed for the reconstruction step. Finally one can note that in the transport algorithm, the procedure Update_bound generating ghost cells for the flux variables has not the same type of arguments as the other procedures _bound. This comes from the fact that the ghost cells for the flux can not be extracted and are computed from the different ghost cells previously created, Spray_bound, Mean_bound, Slope_bound and Recons_bound.

6.5 Code coupling for gas-liquid interactions

A natural evolution for the multi-fluid solver is to be coupled with a gas solver to realize two phase flow simulations. Therefore we need to define a way for both solvers to exchange the needed information. In order to allow an easy implementation of the coupling in a gas solver and to preserve independence between the gas and the liquid parts, we decide to construct a multi-fluid library to use in the gas solver. The choices made for the coupling in a one-way framework are illustrated here, as used for the simulations of Part. IV. There is no limitation to extend to two-way framework, but time evolution algorithms of the gas and the Eulerian liquid phase have to be slightly modified, as explained in Chpt. 8, where the two-way extension, in the framework of the coupling with the CORIA's solver ASPHODELE, developed by J. Reveillon and co-workers, is discussed.

Fortran 14 Fortran implementation for ghost cells generation

```
subroutine Recons_Bound(Bound_spray,Spray)
!
type(comp_field4d), intent(inout) :: Bound_spray
type(field_5d), intent(in)      :: Spray
!
!-local variables
!n(Ndim,bound_number,current_pos,opposite_pos)
integer(kind=ip), dimension(4_ip) ::  n
integer(kind=ip), dimension(4_ip) ::  bound_disc !
integer(kind=ip) ::  i, j
!
!-Number of points for the boundaries in bound_disc :
!-x (r in 2D axi) bound_disc(:) = (/ Ny, Nz /)
!-y (z in 2D axi) bound_disc(:) = (/ Nx, Nz /)
!-z               bound_disc(:) = (/ Nx, Ny /)
!-Obtained in the loop excluding current direction i from grid%disc(Nx,Ny,Nz)
!-
!-To use routine fill_bound_spray we need n (last arg) :
!-  n(:)  = (/Ndim,bound_number,current_pos,opposite_pos/)
!
!- INLET/OUTLET  BOUNDARIES
do i=1_ip,Grid%Ndim
  !-General operation================================================
  !-excluding current direction from grid%disc with pack f90 function
  bound_disc(1_ip:2_ip) = pack(grid%deb(:,my_rank),mask=(/(j/=i,j=1_ip,3_ip)/))
  bound_disc(3_ip:4_ip) = pack(grid%fin(:,my_rank),mask=(/(j/=i,j=1_ip,3_ip)/))
  !
  !-INLET CONDITION==================================================
  n = (/ i, 2_ip*i-1_ip, grid%deb(i,my_rank), grid%fin(i,my_rank) /)
  call fill_bound_spray(Bound_spray%c(2_ip*i-1_ip),Spray,bound_disc,n)
  !
  !-OUTLET CONDITION=================================================
  n = (/ i, 2_ip*i, grid%fin(i,my_rank), grid%deb(i,my_rank) /)
  call fill_bound_spray(Bound_spray%c(2_ip*i),Spray,bound_disc,n)
  !
enddo
!
end subroutine Recons_bound
```

6.5.1 Solver communications

Three types of variables are to be transmitted from the gas solver to the eulerian liquid solver in the studied framework.

We need first to communicate to the liquid solver all the physical constants of the simulation, as the complete description of the mixture at infinite, done in the gas phase solver. This communication is done once at the beginning of the computation.

Then we need to convey the needed gas variables, as density, velocity, diffusivity, heat capacity, etc. The required data depend on the type of models used in the liquid computation. These exchanges have to be done throughout the whole computation. They are needed for the phase space evolution of the spray, for example the gas velocity is used for the drag force. When using Radau, we need to provide, for the procedure `fed` computing the right-hand side of the system of ODE, see Sec. 6.3.3, the expression of the right-hand side as a function of time. The gas variables may then be transfered for several timestep : $n+1$, n, $n-1$, $n-2$..., to interpolate the gas properties at an arbitrary time t, as needed in the Radau5 program.

Finally, the third type of exchanges concerns space grid and timestep. As far as space grid is concerned, we couple two different types of methods using different types of mesh. Indeed, the gas scheme is based on finite difference methods at the cell edges, while the eulerian multi-fluid method uses a finite volume scheme at the cell centers. Furthermore, we might need different mesh refinements for gas, using high order schemes, and the multi-fluid using second order numerical scheme, see Sec. 3.2. We would then need to spatially interpolate the gas fields at the points of the multi-fluid grid. As we are considering non evolving grids, the interpolation coefficients can be pre-calculated. Nevertheless, if we want to use the same number of points for the gas and the liquid, we construct the gas and liquid grids so that the gas grid edges correspond to the liquid cell centers, and that we do not need any grid interpolation. Concerning the timestep, we also have different constraints for the gas and the liquid, where we have a unity-CFL. We then want to manage two different time steps in the coupling, and to solve several gas timestep for one liquid timestep.

6.5.2 Implementation: Eulerian multi-fluid library

The genericity set up in the Eulerian multi-fluid solver allows to define a procedure computing a multi-fluid model timestep evolution of a given spray variable, given the gas variables and physical constants:

procedure eulerian_timestep(Spray,Gas_array,thd)

where Gas_array is an array of `gas_type` variables, containing the solved gas properties:

```
type gas_type
real(kind=rp), dimension(:,:,:),allocatable    ::  Rho
real(kind=rp), dimension(:,:,:),allocatable    ::  Mu
real(kind=rp), dimension(:,:,:),allocatable    ::  Lambda
real(kind=rp), dimension(:,:,:),allocatable    ::  D
real(kind=rp), dimension(:,:,:),allocatable    ::  Cpf
real(kind=rp), dimension(:,:,:),allocatable    ::  Cpg
```

```
real(kind=rp), dimension(:,:,:),allocatable    ::  Tpt
real(kind=rp), dimension(:,:,:),allocatable    ::  Pres
real(kind=rp), dimension(:,:,:,:), allocatable ::  U, Y
real(kind=rp), dimension(:,:,:,:), allocatable ::  Ycf
end type gas_type
```

We use an array of this type in order to possibly pass the gas properties for several timestep for interpolation purpose, as explained in Sec. 6.5.1. The argument `thd` is a derived data type with the thermodynamical constants of the problem. The idea is to provide a kind a eulerian multi-fluid library, that can be used by other codes.

In order to access efficiently to the eulerian_timestep procedure, we create a module, `euler_timevol`, associated with all the multi-fluid library procedures needed by the eulerian_timestep procedure. The idea is to develop two tools using this module. First a multi-fluid solver without gas resolution, allowing to perform fine characterizations of the method using simple analytical gas flows or data from experiments, as done in Part. II. Second a two-phase flow solver, obtained extending a gaseous solver with the multi-fluid library, allowing to perform computations in more complex cases, as done in Part. IV. These two tools are using the same multi-fluid procedures, provided by the module `eulerian_timevol`. These tools evolve then automatically when the multi-fluid library evolves.

Regarding the use of the multi-fluid library in a gaseous solver, we can not directly call the procedure eulerian_timestep, since it needs, as input arguments, data stored in data types provided by multi-fluid library. We thus define a module realizing the interface between the gaseous solver and the multi-fluid library. This module is associated to the library module `euler_timevol`, using a private attribute to avoid interactions between the gaseous solver and the library. We implement in this interface module, `em2c_euler`, generic procedures adapted to the gaseous solver and using the multi-fluid library. These generic procedures are called within the gaseous solver. We write such procedures to realize read/write features for the spray variables, initialization, data transfer for grid and physical constants from the gaseous flow, and call to an interface for multi-fluid timestep evolution, em2c_euler_timestep. The first use of this interface procedure, managing the eulerian timestep resolution, is to store the gas data in appropriate data types, and to manage the storage of this data over several timestep for time interpolation purpose. In this way, the variable Gas_array, can be written, for a p order of interpolation, and for iteration n:

$$Gas\_array = (Gas(i))_{i\in[n-p+1,n+1]} \tag{6.16}$$

where $Gas(i)$ contains all the gas data needed for iteration i. For example a third order interpolation would lead for iteration n to:

$$Gas\_array = (Gas(n-2), Gas(n-1), Gas(n), Gas(n+1)) \tag{6.17}$$

To achieve this storage in Gas_array, we use Algo. 20. The store_gas procedure stores in the multi-fluid library provided variable gas_array and realizes space interpolation if the grids differ. The em2c_euler_timestep procedure deals with the timestep differences between gas and liquid as well. This procedure is directly called within a gaseous solver timestep, and the treatment of the potentially different timestep is illustrated in Algo. 21.

To realize this coupling, we just need to use, in the gas solver timestep resolution, the module `em2c_euler` and to call the procedure em2c_euler_timestep, and some other procedures

Algorithm 20 Gas storage for interpolation needs

```
procedure em2c_euler_timestep(gasite, spray, ite)
...
if (ite<interp_order) then
  {Store in gas_array(i) i=2 to interp_order+1}
  call store_gas(gas_array(ite+1), gasite)
else
  {Shift array : gas_array(i)=gas_array(i+1) and store current ite in gas_array(end)}
  for i=1 to interp_order do
    gas_array(i) = gas_array(i+1)
  end for
  call store_gas(gas_array(interp_order+1), gasite)
end if
...
end procedure em2c_euler_timestep
```

Algorithm 21 Eulerian timestep driving

```
procedure em2c_euler_timestep(gasite, spray, ite, dtg_ite, Tg_ite, thd)
...
call compute_dt_euler(dt_euler, gasite)
...
if (dt_euler<dg_ite or Tg_ite+dt_ite>Telr+dt_euler) then
  ...
  if (dtg > dt_elr) then
    while (Tg_ite+dtg_ite⩾ Telr) do
      call eulerian_timestep(Spray, gas_array, thd)
      Telr = Telr+dt_elr
    end while
  else
    call eulerian_timestep(Spray, gas_array, thd)
  end if
  ...
end if
...
end procedure em2c_euler_timestep
```

as read or write procedures. The eulerian module `em2c_euler` also needs to be used in the initialization of the gaseous solver to call the generic procedure for spray initialization. Very few changes are thus needed to couple a gas solver to the multi-fluid library. Moreover, both parts can evolve independently, the module `em2c_euler` making the coupling evolve if necessary.

Chapter 7

Optimization and numerical efficiency

Contents

Academical research solvers provide a first evaluation of numerical models, prior to their use in industrial scale computations. Therefore, reasonably realistic configurations have to be computed. Besides, given the latest advances in High Performance Computing (HPC) techniques for industrial scale configurations, the out-breaking resolution methods may be associated with an important intrinsic cost. Furthermore, academical solvers must enable to assess the computational efficiency of a given method. Given the importance taken by massively parallel computations, academical solvers must be able to predict the method efficiency in a such context. Academical solvers need thus to be optimized using the different HPC techniques existing at present.
In this work, as we assess the relevance of the multi-fluid method for industrial applications, the developed solver, MUSES3D, has to be optimized through HPC techniques for two reasons.

- First, computational efficiency is needed to assess the more realistic configurations studied in Part. IV.
- Second, the efficiency of the multi-fluid method in a parallel framework is an essential part of its relevance evaluation.

Indeed, an interesting application of the multi-fluid method for an industrial device computation, would be a massively parallel LES, as done actually with AVBP solver, for instance in (Boileau 2007), for the case of an industrial two-phase burner.

This chapter determines how to optimize the multi-fluid method implementation. It aims at evaluating the efficiency one can reach in this context. This study is essential to establish the feasibility of 3-D multi-fluid computations, that is required in Chpt. 11.

The techniques that can be used for single processor optimization are first studied. Given the actual processor characteristics, the sequential cost comes mainly from memory access. Providing a generic optimization in this context is not an easy task. However some generic elements are provided. Even if this optimization cannot suffice, it is an interesting issue in the multi-fluid framework. Indeed, as shown in Chpt. 6, the arrays used for a generic multi-fluid implementation have a high rank, and therefore can lead to a high cost associated to memory access.
Domain decomposition is realized in the multi-fluid framework, to obtain an efficient parallel method. As the multi-fluid model resolves the size distribution, a 3-D computational domain leads to 4-D computations. The different decomposition strategies available in this context are thus evaluated. A specific attention is devoted to the communication implementation, in order to obtain the higher efficiency, in the framework of a given computing architecture.
This study provides the efficiency and speed-up of a multi-fluid parallel computation.

7.1 Single processor optimization

The eulerian multi-fluid solver for spray, will be used, for example for the 2-D computations, in its sequential version. We therefore want to achieve good performance in a single processor run. Furthermore, single processor efficiency is also of importance for parallel computations. Single processor optimization is not an easy task in our framework for two reasons. First, it is not easy to realize a portable optimization, as processor characteristics may strongly influence it. For example, (Girou and Grasseau 1998), was devoted to T3E Cray processor. Second, many optimization techniques, as loop unrolling or subroutine inlining, destroy the code genericity. Indeed, loop unrolling can difficultly be implemented because nearly all the loop indexes are not known *a priori*. Moreover, subroutine inlining, where we replace the call of the subroutine by its contents, clearly prevent to maintain the code genericity. One has to note that we do not want to produce a frozen optimized version of the code, but to obtain the best level of efficiency with our evolving generic code. As any optimization, we focus on the most computationally intensive portions. In our case, we spend the main part of the computational time in the loops used in physical transport therefore we focus here on memory use and loop optimization. One has to note that the Radau5 program of the phase space transport is also responsible for a large part of the CPU time consumption. Nevertheless, we already find the most efficient implementation of the ODE system, see Sec. 3.2.3, and the Radau5 itself is known as being the most efficient implementation of the Radau IIA scheme.

7.1.1 Cache access

We refer for this part to (Girou and Grasseau 1998) and (Sinkovits 2001), from which we tried to extract general features for single processor optimization, as far as memory is concerned. Cache memory is a small part of the memory that the CPU can access very rapidly. We have generally two cache levels, L1 and L2, with a hierarchy, higher level of hierarchy are faster and smaller. Above the cache levels, we find the registers, where are stored the data directly used by the processor, and where only hundred of bytes can be stored. Below the cache level is the main memory, see the memory hierarchy on Fig. 7.1. At present, the CPU are so fast that the memory transfers are the main limitation on processing speed. It is therefore important to focus on memory access for single processor optimization. First of all, we use in the code few high rank arrays rather than several low rank arrays, see Sec. 6.2, that is an interesting point for cache re-use. Moreover, to improve memory access within the transport scheme loops, we perform loop interchange, i.e, we set data contiguity in the loops. In Fortran, an array `Array(i,j,k)` is stored contiguously in `i`, `j`, `k`, so in order to have contiguity the loop:

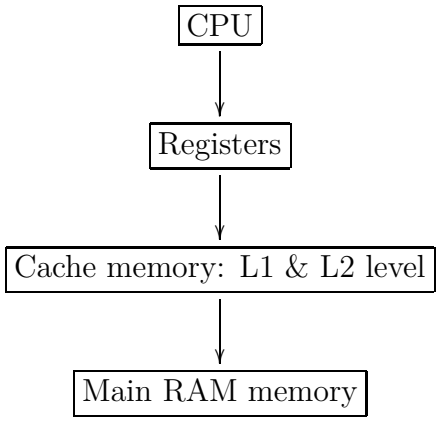


Figure 7.1 – *Memory hierarchy, from the registers to the main memory.*

Fortran 15 Non memory contiguous loop with Fortran storage

```
do i=1,Nx
  do j=1,Ny
    do k=1,Nz
      Array(i,j,k) = ...
    enddo
  enddo
enddo
```

must be interchanged:

Fortran 16 Memory contiguous loop with Fortran storage

```
do k=1,Nz
  do j=1,Ny
    do i=1,Nx
      Array(i,j,k) = ...
    enddo
  enddo
enddo
```

This operation can be done by the compiler but we do it manually because it would have serious performance implications if missed by the compiler. This approach minimizes data stride and allows cache reuse. In our case, we first define the form of our arrays. As explained in Sec. 7.2.1, we choose to cut the domain in physical space for parallel computations. We then set, in order to privilege data contiguity in the parallel computation communications, the following shape for a multi-fluid array : MF_array($N_{\text{mom}}, N_S, N_x, N_y, N_z$). A Fortran loop on this array is then:

Fortran 17 Memory contiguous array cover

```
do k=1,Nz
  do j=1,Ny
    do i=1,Nx
      do s=1,Nx
        do m=1,Nx
          MF_array(m,s,i,j,k) = ...
        enddo
      enddo
    enddo
  enddo
enddo
```

Nevertheless, loop interchange does not resolve all the memory access problem of a numerical scheme loop. Indeed, we need, for example to compute a flux, to access to the neighbors and the loop can have the following structure:

Fortran 18 Problem of spatial locality data

```
do k=1,Nz
  do j=1,Ny
    do i=1,Nx
      do s=1,Nx
        do m=1,Nx
          flux(m,s,i,j,k) = A(m,s,i,j,k) + A(m,s,i-1,j-1,k-1)...
        enddo
      enddo
    enddo
  enddo
enddo
```

In this case we have data stride and the number of elements N_{elements} between `A(m,s,i,j,k)` and `A(m,s,i-1,j-1,k-1)` can be written:

$$N_{\text{elements}} = \underbrace{N_x \times N_y}_{k-1} + \underbrace{N_x}_{j-1} + \underbrace{1}_{i-1} \tag{7.1}$$

To improve efficiency of such loops, one can perform cache blocking, dividing the loop into smaller blocks holding in the cache memory, and adding external loops to control the block size. This procedure can be quite heavy to implement, the loop becoming:

Fortran 19 Cache blocking

```
do kb=1,Nz,n
  do jb=1,Ny,n
    do ib=1,Nx,n
      do k=kb,min(n,kb+Nz+1)
        do j=jb,min(n,jb+Ny+1)
          do i=ib,min(n,ib+Nx+1)
            do s=1,Nx
              do m=1,Nx
                flux(m,s,i,j,k) = A(m,s,i,j,k) + A(m,s,i-1,j-1,k-1)...
              ...
```

where `n` is the number of elements in the block for each direction. We decide, to avoid memory access problems, to create new arrays containing shifted value so that for example `A_shifted(m,s,i,j,k)=A(m,s,i-1,j-1,k-1)`, that simply eliminates data stride. We implement this feature in the code through neighbor array creations for the physical transport scheme.

7.1.2 Neighbor arrays for data space locality

The use of the neighbors mostly appears in the physical transport procedures. In this framework, we use shifted arrays to obtain the neighbors, where we integrate the boundary conditions, described in Sec. 6.4. To integrate boundary conditions in the neighbors, we fulfill the shifted array with the corresponding ghost cells. For example if we shift the two-dimensional array A, of one element column-wise, we fulfill the vector A(:,1) with the vector of ghost cells b:

$$\begin{pmatrix} a_{1,1} & \dots & a_{1,n} \\ \vdots & \vdots & \vdots \\ a_{n,1} & \dots & a_{n,n} \end{pmatrix} \xrightarrow[\text{ghost cells}(b_i)_{(1\leq i\leq n)}]{\text{shift in y},\Delta=+1} \begin{pmatrix} b_1 & \dots & a_{1,n-1} \\ \vdots & \vdots & \vdots \\ b_n & \dots & a_{n,n-1} \end{pmatrix} \tag{7.2}$$

The point we underline here is the use of the intrinsic Fortran function `eoshift` allowing an optimized array shift. The array shifting is achieved only once and it improves several loops within the physical transport algorithm. Furthermore, this technique also has a great interest for genericity, the neighbors containing automatically boundary condition informations, allowing a treatment of boundary conditions fully decoupled from the rest of the transport scheme. Finally we can generate any type of number of neighbors required, and therefore many type of transport numerical schemes.

In order to present the algorithm constructed to realize such integration, we focus here on the case of the flux computation procedure. In this case, we need to construct neighbors for the slopes and the reconstructions, as we see in the general expression of the flux in the 2-D Cartesian dimensionally split case, Eq. (3.55). The neighbor creation used two procedures called in the Flux_evaluation procedure:

Algorithm 22 Flux evaluation procedure: neighbor creation

```
procedure Flux_evaluation(Flux,Recons,Slope,Mean,Spray)
...
{Define the neighbors required for Slope and Recons}
call neighb_def_slope(sfhit_val_slope,shift_dim_slope,type_scheme)
call neighb_def_recons(sfhit_val_recons,shift_dim_recons,type_scheme)
{Construct neighbor arrays}
call neighb_comp(Slope_shift,sfhit_val_slope,shift_dim_slope,Slope)
call neighb_comp(Recons_shift,sfhit_val_recons,shift_dim_recons,Recons)
...
{Flux computation}
call Flux_comp(Flux,Slope,Slope_shift,Recons,Recons_shift)
...
end procedure Flux_evaluation
```

The neighb_def procedures, define the type of neighbor to compute, and therefore depend on the numerical method chosen, indicated by the argument type_scheme. The idea is to define the neighbors needed by the method in terms of shifted directions with their corresponding shift step. This is done defining two arrays: shift_dim and shift_val, used to store, respectively, the directions of the shift and its step. shift_dim is a vector with shape shift_dim(Ndim_transp) and shift_val is a two-dimension array with the shape shift_val(N_neighb,N_dimtransp), where N_neighb is the number of neighbors, and N_dimtransp is equal to one in the case of dimensional splitting, and equal to the number of dimensions in the domain otherwise. We now illustrate, in a 2-D example, the contents of these arrays. To do so, we construct shift_dim and shift_val in order to obtain the five neighbors needed for a 2-D Bouchut flux, see Eq. (3.35):

```
{Dimension to be shifted}
shift_dim(1:2) = (1,2)
{Shift values for the first direction}
shift_val(1:5,1) = (-1,-1,-1,1,1)
{Shift values for the second direction}
shift_val(1:5,2) = (0,-1,1,-1,1)
```

These arrays allow us to construct the neighbors $(a_{i-1,j}, a_{i-1,j-1}, a_{i-1,j+1}, a_{i+1,j-1}, a_{i+1,j+1})$. Therefore, to obtain the neighbor number i, in a 2-D case, we shift the corresponding array of shift_val(i,1) in the first direction, and shift_val(i,2) in the second direction. This operation is done in the second procedure neighb_comp.

In the neighb_comp procedure, we realize the array shift given the arrays shift_dim and shift_val. To do so we use the intrinsic Fortran procedure `eoshift`, allowing to shift an array and to fulfill the resulting shifted array replacing end off boundary elements with a given boundary. The syntax is the following:

```
eoshift(array,shift,boundary,dim)
```

where `shift` if the shift step, `dim` the dimension shifted and `boundary` the padding elements. In order to illustrate the use of this function, we give here the subroutine `neighb_comp`, in the example of six-dimension arrays as Slope and Recons. In this subroutine, we loop

over the number of neighbors (i), and, for one neighbor, we loop over the dimensions of the domain (j), more precisely over Ndim_transp to take into account dimensional splitting. We shift the initial Field in all the directions to obtain the current neighbor Field_shift. In order to pad with the boundary, we use the ghost cells stored in the array Bound%c(2*i) and Bound%c(2i-1) for the frontier i, as defined in Sec. 6.4.2, Eq. (6.14) and Eq. (6.15). For the j direction, we then pad, for the neighbor i, with Bound%c(2*j-1) if the shift value is shift_val(i,j)=-1 or with Bound%c(2*j) if the shift value is shift_val(i,j)=1.

Fortran 20 Intrinsic Fortran functions based neighbor computation -I-

```
subroutine neighb_comp_field6d(Field_shift,Field,Bound,shift_val,shift_dim)
!
type(comp_field6d), intent(inout) ::  Field_shift
type(field_6d), intent(in)        ::  Field
type(comp_field5d), intent(in)    ::  Bound
integer(kind=ip), dimension(:,:),intent(in)::  shift_val
integer(kind=ip), dimension(:),intent(in)  ::  shift_dim
!
!-local variables
integer(kind=ip), dimension(:), allocatable::  shift_pos
integer(kind=ip) ::  Nms, Nmv, Nmsv, Nmt, Ndimt
integer(kind=ip) :: numb_neighb, i, j, k
!
Ndimt = grid%Ndim_transp
Nms   = grid%Nmom_sizes
Nmv   = grid%Nmom_vel
Nmsv  = grid%Nmom_sizes_vel
Nmt   = grid%Nmom_temptherm
numb_neighb = size(shift_val,1_ip)
!
!-Computing shift_pos :
! the number of the array column to shift
! in order to shift in the direction shift_dim
! Array structure Field_shift%array(Nmom,Nd2,Nsec,Nx,Ny,Nz)
!=> if shift_dim==1 shift_pos==4
!=> if shift_dim==2 shift_pos==5
!=> if shift_dim==3 shift_pos==6
!=> shift_pos(i)  = shift_dim(i)+3
!
```

This approach allows to improve cache access in loops, diminishing data stride. Nevertheless, the array shifting procedure have a cost.

We presented here the strategy used to obtain a better efficiency on one processor, preserving the genericity. A more precise profiling of the code should now be done in order to highlight the computational consuming operations. Statement level optimization, e.g, replacing extensive operations by less expensive ones, as replacing power function by separate calls to logarithm and exponential functions, could also be conducted if needed. Nevertheless, the main code optimization is done parallelizing the code.

Fortran 21 Intrinsic Fortran functions based neighbor computation -II-

```
allocate(shift_pos(numb_neighb))
shift_pos(:)  = shift_dim(:)+3_ip
!
do i=1_ip,numb_neighb
  Field_shift%c(i)  = Field
  do j=1_ip, Ndimt
    !-Test i+1 or 1-1 to compute the number of the bound
    !-Take care of dimensional splitting : j=1 but we want
    !-the bound corresponding to the effective direction :
    !=> effect_dim = 1 i-1 : Bound(1); i+1 : Bound(2)
    !=> effect_dim = 2 i-1 : Bound(3); i+1 : Bound(4)
    !=> effect_dim = 3 i-1 : Bound(5); i+1 : Bound(6)
    !-use effect_dim(j)
    if (shift_val(i,j)==-1_ip) then !i-1
      k = 2_ip*grid%effect_dim(j)-1_ip
    elseif (shift_val(i,j)==1_ip) then !i+1
      k = 2_ip*grid%effect_dim(j)
    elseif (shift_val(i,j)/=0_ip) then
      stop 'error shift_val : value possible (-1,1,0) only'
    endif
    if (shift_val(i,j)/=0_ip) then
      !-SIZE MOMENTS=====================================================
      if (Nms>0_ip) &
      Field_shift%c(i)%sizes = eoshift(Field_shift%c(i)%sizes, &
                                       shift_val(i,j),Bound%c(k)%sizes, &
                                       shift_pos(j))
      !-VELOCITY MOMENTS=================================================
      if (Nmv>0_ip) &
      Field_shift%c(i)%vel = eoshift(Field_shift%c(i)%vel, &
                                     shift_val(i,j),Bound%c(k)%vel,&
                                     shift_pos(j))
      !-SIZE-VELOCITY CROSSED MOMENTS====================================
      if (Nmsv>0_ip)&
      Field_shift%c(i)%sizes_vel = eoshift(Field_shift%c(i)%sizes_vel,&
                                           shift_val(i,j), &
                                           Bound%c(k)%sizes_vel, &
                                           shift_pos(j))
      !-THERMODYNAMIC TEMPERATURE MOMENTS================================
      if (Nmt>0_ip) &
      Field_shift%c(i)%temptherm = eoshift(Field_shift%c(i)%temptherm,&
                                           shift_val(i,j), &
                                           Bound%c(k)%temptherm, &
                                           shift_pos(j))
    endif
  enddo
enddo
deallocate(shift_pos)
end subroutine neighb_comp_field6d
```

7.2 Parallel computation in multi-fluid framework

As for standard eulerian computational fluid dynamics, domain decomposition appears, for multi-fluid computation, as a very interesting way to achieve parallel computation. Indeed it offers the ability to use an arbitrary high number of points, the number of process used allowing to have a sub-domain on each process with a reasonable number of points, leading to reasonable computational time and memory requirement. The difficulty of such parallelization lies in the data communications between the involved processes.

7.2.1 Domain decomposition strategies for a 4-D problem

The main peculiarity of the multi-fluid, when dealing with domain decomposition, is the size discretization leading to an extra dimension of the problem. This dimension typically contains five to twenty sections for "classical" multi-fluid model and one has to note that recently developed moment techniques for the size phase space, see Chpt. 2, provide a high order in size numerical method leading to a diminution of the number of sections needed. Furthermore, the operator splitting used for the numerical scheme, see Sec. 3.2.2, makes two blocks appear with different properties, as far as domain decomposition is concerned. On the one hand the physical space transport is local in size and would naturally lead to a domain decomposition in size, each process realizing the transport for one size section. In this case no communication would be necessary. On the other hand, the phase space transport is local in physical space and would then lead to a decomposition in space, a process treating a space sub-domain with all the size sections on it. Here again, no communication needs to take place. The way to decompose the domain is thus not straightforward and we have study the different strategies conceivable to find the best compromise for the two blocks. We can sum up this block structure:

- Physical space transport : local in size
  $\Rightarrow$ no communication for size domain decomposition;
- Phase space transport : local in space
  $\Rightarrow$ no communication for spatial domain decomposition.

We thus envisage three types of domain decomposition:

- a physical space privileging decomposition;
- a phase space privileging decomposition;
- an hybrid decomposition adapted to both parts.

For the first strategy, as we already said no communication will take place for the physical space transport. On the other hand, during the phase space transport, a process will need the information from other process to know neighbor sections at the grid point considered. Nevertheless, this strategy is limited by the number of section used. Indeed it is about five to twenty, and therefore this strategy will have to be coupled with a partially spatial decomposition, in order to use an important number of parallel processes.

If we privilege phase space transport, the communications will then take place in the physical transport where data from other processes will be needed when treating the edges of the considered sub-domain.

Finally, we can think of an hybrid method, with size decomposition in the physical transport, and space decomposition in the size phase space transport. In this case, all the communications are done between the physical and phase space resolution. The array reorganization needed in this strategy will lead to a high amount of communications.

7.2.2 3-D test case and strategies evaluation

In order to evaluate these different strategies and choose one of them, we design a 3-D test case, i.e., 4-D with the size direction, and realize domain decomposition with the different strategies in order to estimate their efficiency. We thus design a very simple test case with the following steps:

Algorithm 23 Domain decomposition test case

```
while (t<t_end) do
  {Simulate physical transport communications}
  call phystransp_mpi_comm(B,C)
  {Simulate Physical transport algorithm}
  call phystransp(A,B,C)
  {Simulate phase transport communications}
  call phasetransp_mpi_comm(B,C)
  {Simulate phase transport algorithm}
  call phasetransp(A,B,C)
end while
```

where A,B,C are five-dimension arrays, A($N_x, N_y, N_z, N_S, N_{\text{mom}}$), are they are supposed to contain several moments of the NDF. In the phystransp and the phasetransp procedures, we very simply simulate the behavior of the physical and phase space transports, manipulating arrays, locally in size for physical transport and locally in space for phase space transport. Basically the idea of both procedures can be summarized as follows:

Algorithm 24 Test case procedure for physical space transport

```
procedure phystransp(A,B,C)
...
for m=M1 to M2 do
  for s=S1 to S2 do
    for k=K1 to K2 do
      for j=J1 to J2 do
        for i=I1 to I2 do
          A(i,j,k,s,m) = B(i-1,j,k,s,m)+B(i,j-1,k,s,m)+B(i,j,k-1,s,m) +
          C(i+1,j,k,s,m)+C(i,j+1,k,s,m)+C(i,j,k+1,s,m)
        end for
      end for
    end for
  end for
end for
...
end procedure phystransp
```

and:

Algorithm 25 Test case procedure for phase space transport

```
procedure phasetransp(A,B,C)
...
for m=M1 to M2 do
  for s=S1 to S2 do
    for k=K1 to K2 do
      for j=J1 to J2 do
        for i=I1 to I2 do
          A(i,j,k,s,m) = B(i,j,k,s-1,m)+B(i,j,k,s,m)+B(i,j,k,s+1,m) + C(i,j,k,s-
          1,m)+C(i,j,k,s,m)+C(i,j,k,s+1,m)
        end for
      end for
    end for
  end for
end for
...
end procedure phasetransp
```

where the beginning and ending of the loops depend on the domain decomposition strategy. On the other hand, the procedures phystransp_mpi_comm and phasetransp_mpi_comm, realize the mpi_communication for the defined strategy. The idea here is not to reproduce the cost of the physical and phase space transports but to analyze the influence of the domain decomposition strategy on the communication costs.

The first strategy, avoiding communication for the physical space transport resolution, normally consists in cutting up the domain in size. Nevertheless, as we have at most 20

sections, this choice is too restrictive. We thus decide to cut up in size and then in z direction, see Fig. 7.2. This strategy finally lead to communication in both physical and

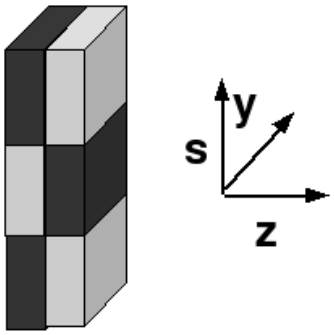


Figure 7.2 – *Representation of the S-Z domain decomposition in size and z direction. The alternatively black and white blocks illustrate the sub-domain of the different processes.*

phase space transport. Furthermore it does not give enough flexibility for the choice of the number of process that must be proportional to the number of size sections. We refer to this decomposition by $S - Z$ afterward.

In order to realize the phase-space privileging decomposition, we cut the domain following the z space direction, see Fig. 7.3. In this case the exchanges are simple and of one single

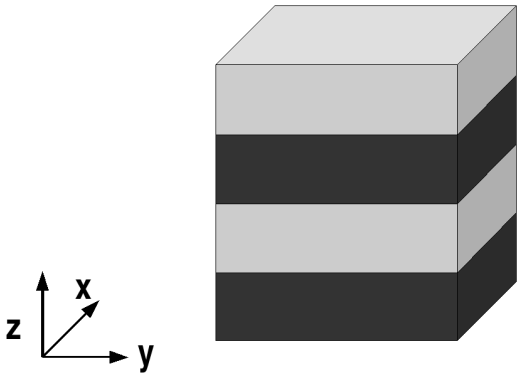


Figure 7.3 – *Representation of the Z domain decomposition in z-direction. The alternatively black and white blocks illustrate the sub-domain of the different processes.*

type. Nevertheless there is a high volume of data for each exchange. This decomposition is called Z afterward. When cutting the domain in space, we can also cut into two directions. We represent in Fig. 7.4 a decomposition cutting up along the z and y directions. This decomposition needs two types of communications, but with less data volume than the one only along z. We will refer to this decomposition as $Z - Y$ decomposition.

Finally we can define a decomposition combining the ones previously describes. We combine a cut along size and z directions for the physical transport and a cut only along z for the

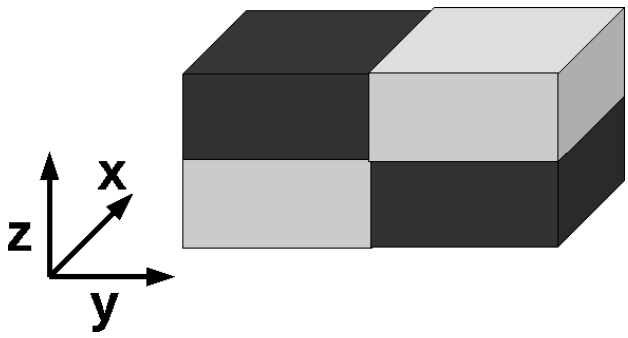


Figure 7.4 – *Representation of the Z-Y domain decomposition in z- and y-directions. The alternatively black and white blocks illustrate the sub-domain of the different processes.*

phase space transport, see Fig. 7.5. This decomposition is called $S - Z/Z$ in the following. Nevertheless, even if this strategy has the advantage to be straightforwardly extended from

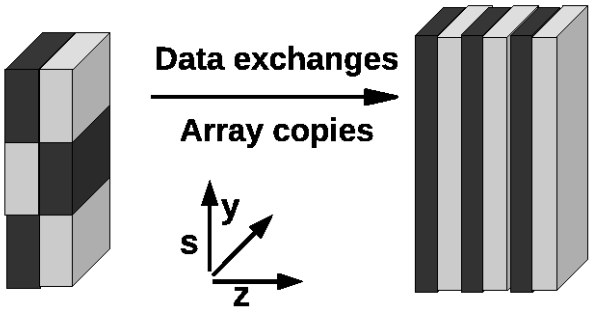


Figure 7.5 – *Representation of the SZ-Z decomposition alternating size-space decomposition and space decomposition. The alternatively black and white blocks illustrate the sub-domain of the different processes.*

of the scalar case, regrouping all the communications between the physical and phase space scalar procedures, we do not expect it to be very efficient, the communications needed to pass from one decomposition to another being huge.

In order to assess these strategies, we compute our 3-D test case implemented with each strategy to compare the computational time. The computations were done in the cluster of the EM2C laboratory, composed of 204 AMD Opteron cores, with 124 cores at 2.4 GHz and 80 cores at 2.2 GHz and an InfiniBand connexion. For the test case, we used 60 2.2Ghz cores with InfiniBand network. Three different grids were used. For time measurements, two types of measure are used to have a reliable value. External time measures, using OS provided tools as `clock()` or `time()` for Linux-type systems, are firstly done. They are compared to internal measures, obtained from instructions inside the code, using for example `cpu_time` procedure in Fortran90/95. It ensures to measure both the computing and the communication times that is often not included in the result provided by internal code measures. No significant differences are to be found between these two measures in our case, therefore we give the internal measure. The details are summarized in Table 7.1. These results allow to draw three conclusions. First, as far as physical space decomposition

Computational Domain	number of process	$S-Z$	Z	$Z-Y$	$S-Z/Z$
10×250^3	60	154	148	**93**	934
10×125^3	60	22	21	**12**	287
10×125.250^2	60	80	74	**47**	573

(7.3)

Table 7.1 – *Cpu time (s) for the different domain decomposition strategies, computed with 60 processors with the 3-D test case. The decomposition along two space directions, $Z-Y$ always give better results.*

	Z	$Z-Y$
Number of communications	3	4
Number of points per communication	$N \times N$	$N \times N/2$
Total number points	$3 \times N^2$	$2 \times N^2$

(7.4)

Table 7.2 – *Evaluation of the number of communications and their number of elements, for a cubic domain with N^3 points, decomposed in four subdomains with the strategies Z, see Fig. 7.3, and $Z-Y$, see Fig. 7.4.*

is concerned, the strategy $Z-Y$ is more efficient than the strategy Z. To explain this result, the number of communications and the number of points to be transfered in these two cases are provided in Table 7.2. To compute these values, we take the example of a domain decomposed into four subdomains with the two strategies, as in Fig. 7.3 and Fig. 7.4. Furthermore, the domain is assumed to be cubic, with N points on each side. Although the strategy $Z-Y$ has more communications than the Z one, it is faster because there are less points to transfer. Furthermore the cost of the commnunications is reduced in the InfiniBand network used for the tests. This is another element in favour of the $Z-Y$ strategy. The second conclusion concerns the inefficiency of the decomposition in size, $S-Z$ in Table 7.1. This is due to the fact that this decomposition can not be done only in size given the weak number of points in this direction and thus must be coupled with a physical space decomposition. It thus leads to communications in both part of the splitting. Finally, the idea of using two types of decomposition can not be envision. Indeed, as we said before, the high number of communications needed to switch from one decomposition to another has a prohibitive cost, as can easily be seen in Table 7.1 for the decomposition $S-Z/Z$. We therefore retain a decomposition of the physical space, using the maximum space directions compatible with the number of processes. Indeed, as far as it is possible, we should use a decomposition in the three directions $Z-Y-X$, in order to minimize the number of points to transfer. It maximizes the communication number but it is not critical, especially with an InfiniBand network.

7.3 Process communications

We take advantage of the code's structure, and particularly of the decoupled neighbor treatment, to introduce a modular domain decomposition in MUSES3D. The process communications are based on a Cartesian MPI Topology and are designed for an InfiniBand network, as the one available on the EM2C cluster.

7.3.1 Modular communication introduction

The neighbor handling of MUSES3D, described in Sec. 7.1.2, is also of great interest regarding domain decomposition implementation. Indeed, it enables to keep clear of any communications within the transport scheme, and thus it decouples the communications from the numerical schemes, offering then a high genericity. The communications are wholly contained within the boundary conditions procedures, described for the scalar case in Sec. 6.4. The domain decomposition is introduced in the ghost cells creation procedures. The idea is now to choose if the considered frontier is a bound of the computational domain, leading to the classical ghost cells creation, or a bound between two processes, leading to a communication. The Step_Bound procedure, described in Sec. 6.4.2, becomes:

Algorithm 26 Boundary treatment step_bound procedure

```
procedure Step_bound(Bound_Var,Var,type_bound)
...
for i=1 to N_dim do
  {Generate inlet condition}
  if (type_bound(2i-1)==0) then
    {Communications}
    call send_recv(Bound_Var(2i-1),Var,type_bound(2i-1),2i-1)
  else
    {Classical ghost cell creation}
    call fill_bound_var(Bound_Var(2i-1),Var,type_bound(2i-1),2i-1)
  end if
  {Generate outlet condition}
  if (type_bound(2i)==0) then
    call send_recv(Bound_Var(2i),Var,type_bound(2i),2i)
  else
    call fill_bound_var(Bound_Var(2i),Var,type_bound(2i),2i)
  end if
end for
...
end procedure Step_bound
```

where we set type_bound(i)=0 when the frontier is between two sub-domains and type_bound>0 for the different types of boundary conditions for the computational domain frontiers. The integer n_bound indicates the number of the frontier from one to $2N_{\text{dim}}$, see Eq. (6.15). Nevertheless, communications may also arise at a frontier of the

domain for periodic boundary conditions: the ghost cells of a periodic boundary condition at $x = 0$ are to be taken from the field at $x = X_{\max}$, possibly on another process. Actually communications are needed for periodic conditions in any case except if the domain decomposition is such that the direction of the boundary is not cut off. In this case the process does not have any neighbor in that direction. As we will detail on the next paragraph, the use of MPI topology allows us to know the rank of the preceding and following processes, we can thus check if communications are needed or not for the boundary condition, if the current process rank is respectively different or equal to the one of the neighbor for the considered boundary. The procedure Step_bound, can therefore be written:

Algorithm 27 Step_bound procedure including domain decomposition

```
procedure Step_bound(Bound_Var,Var,type_bound)
...
for i=1 to N_dim do
  {Generate inlet condition}
  if (type_bound(2i-1)==0 or (type_bound(2i-1)==1 and myrank/=mpi_neighb(2i-
  1) ) then
    {Communications}
    call send_recv(Bound_Var(2i-1),Var,type_bound(2i-1),nb_bound)
  else
    {Classical ghost cell creation}
    call fill_bound_var(Bound_Var(2i-1),Var,type_bound(2i-1),nb_bound)
  end if
  {Generate outlet condition}
  if (type_bound(2i)==0 or (type_bound(2i)==1 and myrank/=mpi_neighb(2i) )
  then
    call send_recv(Bound_Var(2i),Var,type_bound(2i),nb_bound)
  else
    call fill_bound_var(Bound_Var(2i),Var,type_bound(2i))
  end if
end for
...
end procedure Step_bound
```

where type_bound(i)=1 corresponds to a periodic boundary condition.

This is the only modifications needed by the domain within the physical transport algorithm, described in Sec. 6.3.2. The domain decomposition strategies kept, see Sec. 7.2.1, have no impact on the phase space transport, since all the size sections at a considered grid point are on the same process. We also modified the allocation and initialization procedures, to deal with the current sub-domain, and we used integer variables for the first and last loop index values.

The structure of MUSES3D thus allowed us to introduce easily the domain decomposition with few interactions with the transport algorithm. The main procedures of the domain decomposition, send_recv and its dependences are isolated in the class `mpi_class`. The entire structure of the communications, can then be modified without modifying the main code.

7.3.2 MPI implementation

We give here some elements concerning the MPI tools we used for the communications implementation, detailing the topology used, the structure of the communication procedures and the type of MPI communications used. For more details about the MPI standard, see (Chergui, Dupays, Girou, Requena, and Wautelet 2006), (Chergui, Dupays, Girou, and Lavallée 2006) and references therein.

Cartesian topology

Given that we use Cartesian regular grid, we naturally use the MPI Cartesian topology. MPI topologies organize the different processes to fit the computational grid; they are very well suited for domain decomposition applications. In Cartesian topology, each process is defined by its coordinates within the process grid, see Fig. 7.6. The process grid can be periodic. We use mainly four MPI topology subroutines:

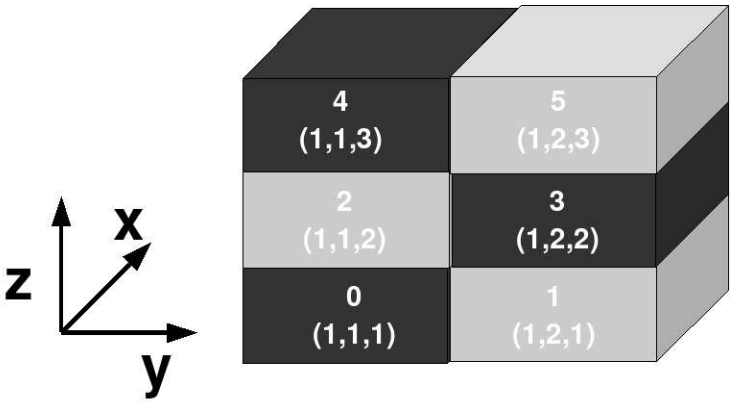


Figure 7.6 – *Example of a 3-D cartesian topology. Each process i can be addressed by its coordinates (n_x, n_y, n_z) within the grid.*

- `mpi_dims_create`
- `mpi_cart_create`
- `mpi_cart_shift`
- `mpi_cart_coords`

The procedure `mpi_dims_create` defines the number of processes in each space dimension given the total number of processes. This function, returns a vector with the number of processes in each direction, used afterward as an input of `mpi_cart_create` creating the Cartesian process grid. Another very interesting feature of the MPI Cartesian topology is the possibility to have the rank of the neighbor processes in each direction with `mpi_cart_shift`. This feature is used for all the communications to define the addressed processes. Finally, `mpi_cart_coords` returns the coordinates of the current process in the grid, also used to define the communications and to construct the computational grid of the current process. See, for example, (Chergui, Dupays, Girou, Requena, and Wautelet 2006) for more informations about Cartesian topology.

Communication structure

The main communication procedure, send_recv, introduced in Sec. 7.3.1, is actually a generic interface for three procedures:

- send_recv_field4d,
- send_recv_field5d,
- send_recv_field6d,

treating data transfer for all the main types of the code. We give here the main steps of one of this procedure, sed_recv_field5d, to illustrate the structure of the communications, in our particular multi-fluid linked array ordering. We recall that five-dimension arrays have the following shape: Array($N_{\text{mom}}, N_S, N_x, N_y, N_z$), see Sec. 6.2 for details. Since the domain decomposition is based on the physical space, see Sec. 7.2.1, we loop over the two first directions of the array to transfer the data for a given physical sub-domain. Moreover, we define three different procedures realizing the transfer for 1-D to 3-D configurations.

Algorithm 28 Parallel data communication procedure

```
procedure send_recv_field5d(mp,Field,Field_extract,nb_bound)
...
if (N_dim == 1) then
  for j=1 to N_S do
    for i=1 to N_mom do
      call send_recv1d(mp,Field(i,j,:,1,1),Field_extract(i,j,1,1),n_bound)
    end for
  end for
else if (N_dim == 2) then
  for j=1 to N_S do
    for i=1 to N_mom do
      call send_recv2d(mp,Field(i,j,:,:,1),Field_extract(i,j,:,1),n_bound)
    end for
  end for
else if (N_dim == 3) then
  for j=1 to N_S do
    for i=1 to N_mom do
      call send_recv3d(mp,Field(i,j,:,:,:),Field_extract(i,j,:,:),n_bound)
    end for
  end for
end if
...
end procedure send_recv_field5d
```

where we pass in the procedures send_recvnd all the physical space points for a given multi-fluid spray moment and size section, Field(i,j,:,:,:) in 3-D. The input argument mp is a derived type containing all the informations needed in the MPI Cartesian topology: rank, coordinates, and grid of the current process, ranks of the neighbor process, etc. The argument n_bound specifies the current space direction and position of the considered

frontier, see Sec. 7.1.2. This flag determines how the data are transfered. Indeed, in a 3-D case, given Eq. (6.15), the communications are organized as follow:

Algorithm 29 Communications for domain boundaries

```
if (n_bound==1) then
  send to East and receive from West
else if  (n_bound==2) then
  send to West and receive from East
else if (n_bound==3) then
  send to Norht and receive from South
else if (n_bound==4) then
  send to South and receive from North
else if (n_bound==5) then
  send to Front and receive from Back
else if  (n_bound==6) then
  send to Back and receive from Front
end if
```

where we define, if we set (i, j, k) the index of processes in the Cartesian process grid (x, y, z), Fig. 7.7:

- East $\Rightarrow\ i+1$, West $\Rightarrow\ i-1$
- North $\Rightarrow\ j+1$, South $\Rightarrow\ j-1$
- Front $\Rightarrow\ k+1$, Back $\Rightarrow\ k-1$

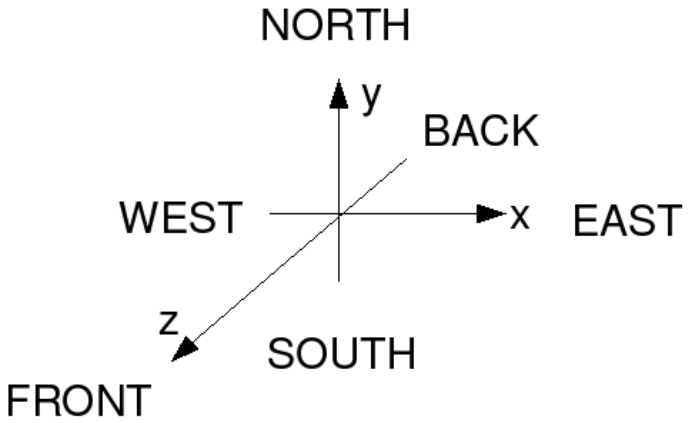


Figure 7.7 – *3-D spatial organization used to describe communications.*

Communication types

To achieve this data transfer, we use point to point communications between current process and its neighbors. We choose synchronous blocking MPI communications. In the synchronous mode, the message sending ends only if receipt is completed, the sending is thus coupled to the receipt. This mode avoids message temporary copying, that can occur

in standard mode when the sending ends before receipt occurs, and its associated cost. We use blocking communications, i.e the code waits the transfer to be completed before going on, for we can not optimize the communications by covering them with computations, the results of the communications being needed for the continuation of the computations. The MPI subroutines for this type of communications are `mpi_ssend` and `mpi_recv`. Moreover, we realize the transfer using a MPI derived data type. We have chosen the shape of the arrays so that the communications exchange memory contiguous data, see Sec. 7.2.1. We therefore use the subroutine `mpi_type_contiguous` to construct the derived data types needed. It has the following syntax:

```
integer, intent(in)  ::  nb_elements, data_type
integer, intent(out) ::  derived_data_type, error_code

call mpi_type_contiguous(nb_elements,data_type,derived_data_type,error_code)
```

In a 3-D case, we thus define a vector `plane`, with shape `plane(3)`, to store the derived data types for the three types of plane to be sent, at the frontiers of the three directions:

Fortran 22 MPI initialization

```
bx=mp%beg(1_ip)
ex=mp%end(1_ip)
by=mp%beg(2_ip)
ey=mp%end(2_ip)
bz=mp%beg(3_ip)
ez=mp%end(3_ip)
!
call mpi_type_contiguous((ey-by+1_ip)*(ez-bz+1_ip),itypereal,plane(1_ip),ierr)
call mpi_type_commit(plane(1_ip),ierr)
!
call mpi_type_contiguous((ex-bx+1_ip)*(ez-bz+1_ip),itypereal,plane(2_ip),ierr)
call mpi_type_commit(plane(2_ip),ierr)
!
call mpi_type_contiguous((ex-bx+1_ip)*(ey-by+1_ip),itypereal,plane(3_ip),ierr)
call mpi_type_commit(plane(3_ip),ierr)
```

where `mp%beg(i)` and `mp%end(i)` are the first and last indexes of the current process sub-domain grid for the direction `i`. The procedure `mpi_type_commit` activates the new derived data type. The data type `itypereal` is not a basic MPI data type (`mpi_real`,`mpi_integer`, etc) but is created with the MPI subroutine `mpi_type_create_f90_real`, in order inherit the fortran numerical types defined in the code with `kind` features. This numerical data type inheritance comes from the MPI-2 standard, see (Chergui, Dupays, Girou, and Lavallée 2006). Therefore, if we focus on the first direction frontiers n_bound=1 at $x = X_{\min}$ and n_bound=2 at $x = X_{\max}$ of the current sub-domain, the communications are implemented as written in the following: The process neighbor `E` and `W` are simply the first and second neighbors of the considered process: `mp%neigh(1)` and `mp%neigh(2)`. An interesting way to simplify the communication implementation would be to use the sub-array constructor provided by the MPI-2 standard subroutine `mpi_type_create_subarray`, see (Chergui, Dupays, Girou, and Lavallée 2006). Indeed we could directly perform the communications

Fortran 23 Procedure for 3-D send/receive

```
subroutine send_recv3d(mp,Field,Field_extract,n_bound)
integer(kind=ip),intent(in)  :: n_bound
type(mpi_type),intent(inout) ::  mp
real(kind=rp),dimension(:,:,:),intent(in)  :: Field
real(kind=rp),dimension(:,:),intent(inout) :: Field_extract
...
N = shape(p)
...
!---First direction ---
!--Send to East neighbor and receive from West
if (n_bound == 1_ip) then
...
!-send Field(Xmax,:,:)
  call mpi_ssend(Field(N(1),1,1), 1_ip, plane(1_ip), mp%neigh(E), &
      tag1, mp%comm, ierr)
!-receive in Field_extract(:,:)
  call mpi_recv(Field_extract, 1_ip, plane(1_ip), mp%neigh(W), &
      tag1, mp%comm, status, ierr)
endif
!
!--Send to East neighbor and receive from West
if (bound == 2_ip) then
...
!-send Field(1,:,:)
  call mpi_ssend(Field(1,1,1), 1_ip, plane(1_ip), mp%neigh(W), &
      tag1, mp%comm, ierr)
!-receive in Field_extract(:,:)
  call mpi_recv(Field_extract, 1_ip, plane(1_ip), mp%neigh(E), &
      tag1, mp%comm, status, ierr)
endif
!
...
end subroutine send_recv3d
```

on the complete Field, without defining loops on the two first directions, as done in procedure send_recv_field5d. Nevertheless it would lead to less communications, one instead of $N_{\text{mom}} \times N_S$, but with more data to transfer $N_{\text{mom}} \times N_S \times N1 \times N2$ instead of $N1 \times N2$, where $(N1, N2)$ are (N_x, N_y) or (N_x, N_z) or (N_y, N_z) depending on the considered frontier. Since we want to optimize the communications for an InfiniBand network, this solution could decrease the performances, since InfiniBand networks are more efficient with a high number of light communications.

This communication structure seems then to be adapted to the InfiniBand network we aim at using. A precise study of the scalability and of the efficiency of this MPI implementation is done in the following.

7.4 Multi-fluid parallel efficiency

The domain decomposition for the multi-fluid method has been implemented in the solver MUSES3D, following the strategies chosen in Sec. 7.2. After validating the parallel version of the solver, the speed up obtained is studied. The precise context used for the parallel solver efficiency evaluation is first defined, as well as the Amdahl and Gustafson laws used to present the speed up results.

7.4.1 Parallel computation validation

In order to validate the domain decomposition done for the multi-fluid, we use a 3-D analytical gas phase, given by a vortical flow extended from the 2-D Taylor-Green vortices defined in Chpt. 4. The interest of this gas field is that one does not need to couple the multi-fluid solver with a solver for the gas phase, as it is analytical.
A 100^3 point grid is used. In order to validate the parallel version of the solver, it is compared to the sequential version previously created and assessed on various configurations, Chpt. 9, Chpt. 10, rather than with the parallel solver with a single process. The parallel computation is done on 24 processes, with a number of processes different in each direction in order to have the most generic case. The following process repartition with spatial direction is set: $N_{\text{p}x} = 4$, $N_{\text{p}y} = 3$ and $N_{\text{p}z} = 2$, where $N_{\text{p}i}$ is the number of processes in the i^{th} direction. The sequential and parallel versions are compiled with the same compiler and the same optimization options. The parallel and sequential computations are done on the same AMD Opteron cores.
These test cases lead to the same results, the difference between the sequential and the parallel cases being under the computer precision. The domain decomposition and the associated parallel developments do not have any influence on the results of the solver. This parallel solver will thus be used in a first 3-D multi-fluid spray computation in Chpt. 11.

7.4.2 Tools for parallel efficiency and speed up evaluation

Above all, the strategy used for the evaluation of a parallel application speed up has to be precisely defined.

First of all, the way used to measure the computational time consumed has to be defined. In the results presented in the following:

- two types of measure are used to have a reliable value, external and internal time measures, as defined in Sec. 7.2.2.
- Significant times are measured, i.e. at least ten minutes.
- The measures are always obtained with a first execution, never using a loop on a test case, that could alter the results through cache effects.

Afterward, the measure conditions have to be defined:

- time measures: both external and internal, as previously defined;
- cluster used: pairs of AMD Opteron dual core 64-Bit processors with 2.4Ghz clock rate, 8Go shared memory. The processors communicate through an InfiniBand network.
- Cache size: 1024 KB.
- OS: Red Hat Enterprise Linux (RHEL) 64-Bit distribution, based on a 2.6.9-34 Linux kernel.
- Compiler: PathScale 64-Bit fortran compiler, pathf95.
- Input/Output: no input/output operation is done in the test cases, to obtain only computational time.

To present the speed up of a parallel application, two point of views can be taken:

- Amdahl's law considers that an application as a sequential part, that can not benefit from parallel computing and a part that can be treated in parallel. For example the time spent in sequential case $T(1)$ can be written $T(1) = T_s^a + T_p^a$, where T_s^a is the purely sequential part and T_p^a the part that can be parallelized. The speed up for the Amdahl's law, $S^a(P)$, is defined as the ratio of the time $T(P)$ of the application on P processes by the sequential time $T(1)$:

$$S^a(P) = \frac{T(1)}{T(P)} = \frac{T(1)}{T_s^a + \frac{T_p^a}{P}} = \frac{1}{\frac{T_s^a}{T(1)} + \frac{1}{P}\frac{T_p^a}{T(1)}} \tag{7.5}$$

  Therefore, one can see that the speed up is limited to $1/(T_s^a/T(1))$ when the number of processes go to infinity. Amdahl's law therefore shows the limits of parallelization.

- Nevertheless, another point of view can be adopted, based on more practical arguments. Indeed, one can consider that the idea behind the use of P parallel processes is to achieve more work than with one process in the same time, rather than achieving the same work faster. Therefore, in a time T_0 a task W_0 can be done with one process, and a task W_P with P processes: $T(1, W_0) = T(P, W_P) = T_0$. To define the speed up, the time taken for the task W_P on a single process is assumed to be: $T(1, W_P) = T_s^g(P, W_P) + P \times T_p^g(P, W_P)$, where T_s^g is the sequential part and T_p^g the time obtained with a parallel computation with P processes. In the Gustafson's law,

the sequential part is assumed to be constant. The speed up is thus given by:

$$S^g(P) = \frac{T(1, W_P)}{T(P, W_P)} = \frac{T_s^g(P, W_P)}{T(1, W_P)} + P \times \frac{T_p^g(P, W_P)}{T(1, W_P)} \tag{7.6}$$

Using this definition, the speed up always increase with the number of processes.

These two visions will be adopted to characterize the efficiency of the parallel multi-fluid solver.

7.4.3 Efficiency and speed up of parallel MUSES3D solver

The test cases used to assess the parallel efficiency of MUSES3D solver must be representative of the cases one intends to compute. We thus choose a domain with a 128^3 point grid and with ten multi-fluid sections. This is representative of the mesh used for the droplet laden Homogeneous Isotropic Turbulence computations done in Chpt. 11. An analytical gas field is still use, in order to focus on the cost of the Eulerian multi-fluid liquid phase solver.

First of all, before assessing the speed up of the code, we analyze the influence of the domain decomposition on the computational time. To do so, we perform test cases with 64 processes, with the repartition given in Table 7.3. This study confirms the idea given in

Test case	N_{px}	N_{py}	N_{pz}	Computational time internal measure (s)
Repart_1	4	4	4	**9782**
Repart_2	0	8	8	9883
Repart_3	8	8	0	9958
Repart_4	0	0	64	13502
Repart_5	64	0	0	13913

Table 7.3 – *Repartition of the parallel process in each direction for the different test cased realizes. The variable N_{pi} represents the number of process in the i-direction.*

Sec. 7.2.2: although the decomposition taking an equivalent number of processes in each direction has more communications, it has less points to transfer and it is thus more efficient. Furthermore, the InfiniBand network of the used architecture favors this decomposition strategy. The strategy Repart_1 is thus the more efficient. Besides, a memory access cost is highlighted in these test cases. Indeed, as seen in the pair of test cases Repart_2/Repart_3 and Repart_4/Repart_5, the domain must be first cut in z-direction, then in y-direction and finally in x-direction. This is due to the form chosen for the Fortran arrays: `Array(Nx,Ny,Nz)`. Therefore, in the case Repart_5 for instance, the communications send and receive sub-arrays: `Array(1:Nxpr,:,:)`, where `Nxpr` is the number of points per process in the x-direction. This sub-array contains non local memory data in Fortran, see Sec. 7.1.1. On the other hand, the sub-arrays involved in the case Repart_4, `Array(:,:,1:Nzpr)` are local in memory. It explains the lower cost of case Repart_2 and Repart_4, compared to Repart_3 and Repart_5, respectively.

As far as speed-up results are concerned, three different domain discretizations are considered, from 64^3 to 256^3 grid points. The number of processes used varies from 8 to 124. One can first plot the Amdahl's law, Sec. 7.4.2, for the speed-up obtained in these configurations, Fig. 7.8. In this case, the parallel efficiency is thought in terms of the time decreased

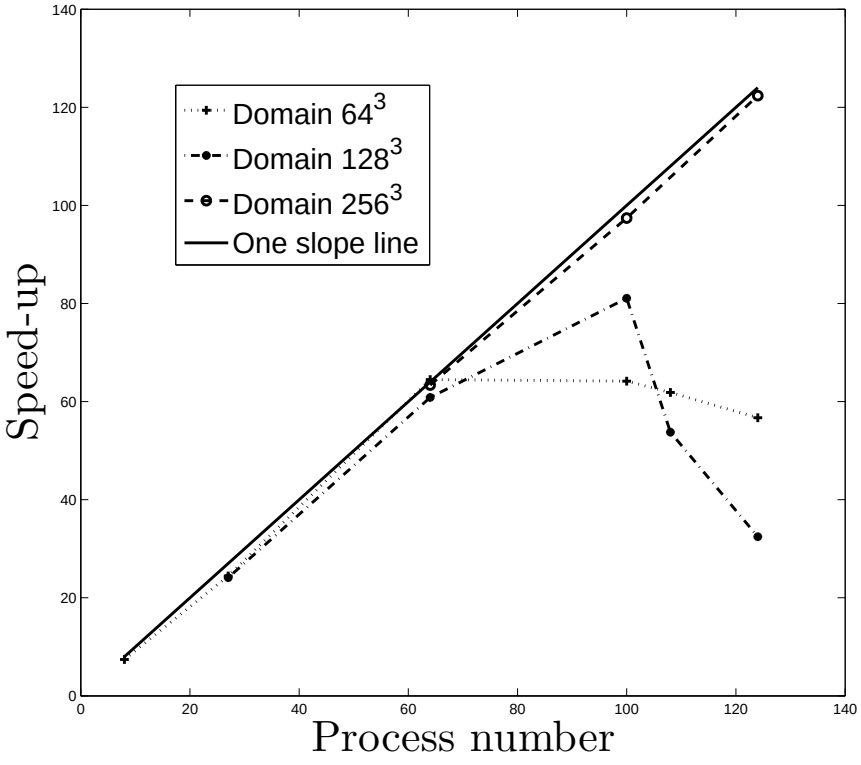


Figure 7.8 – *Speed-up evolution with the number of parallel processes used, for three spatial domain discretization:* 64^3 *(dotted line),* 128^3 *(dashdot line) and* 256^3 *(dashed line). A one slope line is also plotted (solid line). The curve obtained for each discretization corresponds to Amdahl vision. A saturation occurs, except for the* 256^3 *where the number of processes leading to saturation is not reached.*

associated to the number of processes used for a given task. In this context, a saturation occurs where the consumed time cease to decrease while the number of processes still increases. Consumed time can even increase with the number of processes. Figure 7.8 shows in this framework that the maximal parallel efficiency is obtained with 64 and 124 processes for the 64^3 and 128^3 grids, respectively. For the 256^3 domain, the number of processes leading to saturation is not reached. This saturation occurs when the physical domain on a given process is small enough so that time needed by the communications is greater than the individual process computational time. Nevertheless, the parallel efficiency can also be seen from a more practical point of view. Indeed the parallel computation allows to realize, in the same time, a more important task. This represents the Gustafson definition, given in Sec. 7.4.2, illustrated in Fig. 7.9. The Gustafson speed-up is thus constructed with the best speed-up of the different configurations. A unity efficiency is obtained for both 64^3 and 256^3 grids. The maximum efficiency of the 128^3 grid is 0.8.

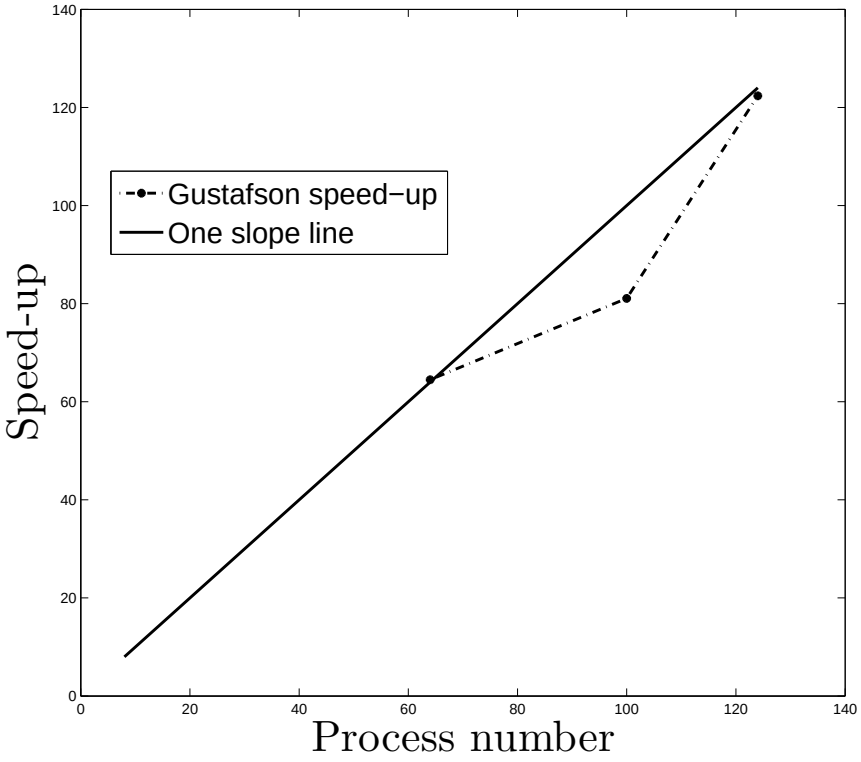


Figure 7.9 – *Speed-up evolution using Gustafson definition (dashdot line). The three points are obtained using the best speed up of the three physical domains:* 64^3*,* 128^3 *and* 256^3*. The* 64^3 *and* 256 *domains have a maximum efficiency of one, whereas the* 128^3 *has a maximum efficiency of 0.8. This Gustafson speed-up definition considers that parallelization allows to execute more work in the same time, rather than the same work within a shorter time. A one slope line is also plotted (solid line).*

This first analysis of the parallel multi-fluid efficiency is thus very encouraging. Indeed, a number of processes equal to the number of points in a direction can be used without reaching saturation, leading to an efficiency between 0.8 and 1, Fig. 7.8. A one slope line is nearly reached in the Gustafson definition, Fig. 7.9, using up to 124 parallel processes. This efficiency allows us to envision 3-D multi-fluid spray computations. A first case of spray dispersion in a 3-D Homogeneous Isotropic Turbulence is assessed in Chpt. 11.

Part IV

Computational multi-fluid dynamics

Introduction

An important advance of this work consists in the achievement of multi-fluid computations in more realistic flows. This progression is needed to assess the applicability of the multi-fluid to industrial scale computations. This study is made possible by the efficient numerical schemes designed, given in Chpt. 3. Furthermore, the encouraging results obtained in the detailed characterization of the multi-fluid method in 2-D vortical flows, Part. II, provide the required basis to envision more realistic flows.
The behavior of the multi-fluid model in more realistic configurations is assessed in this part. This behavior evaluation illustrates:

- the feasibility of such configurations;
- the precision obtained for the dynamics and evaporation of the polydispersed spray;
- the cost associated with the multi-fluid method.

The numerical tools used in this part are described. Given fuel injection computations are the industrial applications of interest behind this study, spray injection in 2-D gaseous jets are tackled in this part. Furthermore, 3-D computations, that represent the achievement of the study, are assessed through the computation of spray dispersion in an Homogeneous Isotropic Turbulence.
The solver resulting of the coupling between MUSES3D solver with the gaseous solver ASPHODELE developed at CORIA, is described in Chpt. 8. It represents a new academical tool devoted to spray model evaluation. It provides both a Eulerian/Lagrangian and a Eulerian/Eulerian descriptions of dispersed two-phase flows. Multi-fluid description of the spray injected in a 2-D axi-symmetrical jet in assessed in Chpt. 9. This evaluation is done through comparisons with a Lagrangian description of the spray. Given the relative complexity of the configuration, the procedure defined and used in Chpt. 4 for Eulerian/Lagrangian comparisons cannot be used. More qualitative comparisons are conducted in this context, albeit allowing to evaluate the time-resolved description of spray size-conditioned dynamics. This configuration is also used to discuss the cost associated to the multi-fluid method. The evaluation of the precision obtained with the multi-fluid method is completed in Chpt. 10, tackling the key issue, for combustion applications, of fuel vapor mass fraction description. First Eulerian/Lagragian comparisons for the fuel vapor are provided. They are based on a new framework for spray method evaluation in combustion context, defined in Chpt. 1. Finally, a 3-D HIT case is achieved in Chpt. 11. It shows the feasibility of 3-D multi-fluid, providing its first achievement. This computation gives the basis for further physical studies of spray dynamics in HIT configuration. Furthermore it provides a first step toward multi-fluid implementation in industrial or semi-industrial computational codes.

Chapter 8

Eulerian and Lagrangian fluid solvers

Contents

In order to assess multi-fluid relevance for industrial devices simulations, spray evolution in more realistic flows is tackled in Chpt. 9, Chpt. 10 and Chpt. 11.
The present chapter introduces the numerical tools used in this framework. Indeed, a new academical solver has been created coupling ASPHODELE solver, developed at CORIA by Julien Reveillon and collaborators, with the multi-fluid solver MUSES3D developed in this study, Part. III. ASPHODELE solver couples a Eulerian description of the gas phase with a Lagrangian description of the spray. The numerical resolution of the gas and liquid phase, and their coupling, as implemented in ASPHODELE, are detailed in the first part of this chapter. The second part is devoted to the coupling achieved between the two solvers. The scientific computing issues concerning the coupling of MUSES3D solver with a gas phase solver were already detailed in Chpt. 6. Attention is devoted in this chapter to Euler/Euler coupling algorithm in the case where fuel evaporates in the gas through FADO procedure, as described in Sec. 1.4.3, and used for Chpt. 10 computations.
Finally, we summarize all the possibilities of this new solver coupling a Eulerian gas description with both Eulerian and Lagrangian descriptions of the spray. It allows for example, in the framework of one-way coupling, to compute simultaneously both spray descriptions with the same gas phase, within the same code run. This solver opens new possibilities of method evaluation in terms of dynamical comparisons and computational cost evaluations.

8.1 Eulerian-Lagrangian gas-liquid solver ASPHODELE

We provide in this section the description of the numerical methods used in the CORIA's solver ASPHODELE, developed by Julien Reveillon and collaborators. This two-phase flow code will provide the gas phase description for both the multi-fluid computations and the Lagrangian description used to assess the multi-fluid behavior. A statistical initialization for the Lagrangian solver, using rejection method given in Chpt. 2, has been implemented in the present study.

8.1.1 Eulerian-Lagrangian phase resolution

The solver ASPHODELE, holds a DNS gas solver for Low Mach or incompressible flows, coupled with a Lagrangian solver, computing DPS or DSMC methods for infinite Knudsen limit, and DPS of colliding configurations.

Eulerian carrier phase numerical scheme

Low Mach number The jets computations presented in Chpt. 9 and Chpt. 10, are conducted, as far as gas carrier phase is concerned, with a DNS solver for low Mach number flows. This solver is based on prediction correction method for the velocity evolution, as was introduced in (Chorin 1968; Temam 1968). For incompressible flows, Eq. (1.106), the evolution of the velocity is done in the following way:

- the momentum conservation equation is solved neglecting pressure effects:

$$(\rho_g \boldsymbol{U_g})^\star - (\rho_g \boldsymbol{U_g})^t = \Delta t \left(-\partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} \otimes \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau})\right); \tag{8.1}$$

- a Poisson equation for the pressure is solved, to compute $\partial_{\boldsymbol{x}}(\tilde{P})$ ensuring the realization of the constraint of the velocity for incompressible flow:

$$\partial_{\boldsymbol{x}} \cdot \boldsymbol{U_g}^{(t+\Delta t)} = 0. \tag{8.2}$$

- Finally, the velocity is corrected solving the momentum equation accounting for the pressure effects:

$$(\rho_g \boldsymbol{U_g})^{(t+\Delta t)} - (\rho_g \boldsymbol{U_g})^\star = \Delta t \left(-\frac{1}{\gamma_\infty M^2}\, \partial_{\boldsymbol{x}}(\tilde{P})\right). \tag{8.3}$$

This kind of technique is described in (Hirsch 1990). Second order method are provided in (Guermond and Shen 2001), and for variable density flow in (Bell and Marcus 1992). The extension for low Mach number flows has been introduced in (Nicoud 2000). In the case of low Mach number flows, the constraint to satisfy for the velocity field is no longer the divergence free constraint but is:

$$\partial_{\boldsymbol{x}} \cdot (\rho_g\ \boldsymbol{U_g}) = -\partial_t\, \rho_g\,. \tag{8.4}$$

The procedure implemented in ASPHODELE is the following:

- Velocity prediction:

$$(\rho_g \boldsymbol{U_g})^\star - (\rho_g \boldsymbol{U_g})^t = \Delta t \left(-\partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} \otimes \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau})\right). \tag{8.5}$$

- Poisson equation with low Mach constraint:

$$\partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g})^{(t+\Delta t)} - \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g})^\star = -\Delta t\ \ \partial^2_{\boldsymbol{x}^2} \cdot (\tilde{P}). \tag{8.6}$$

  From the low Mach constraint, we have:

$$\partial_{\boldsymbol{x}} \cdot (\rho_g\ \boldsymbol{U_g})^{(t+\Delta t)} = -\left(\partial_t\, \rho_g\right)^{(t+\Delta t)}, \tag{8.7}$$

  and from the prediction step:

$$\partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g})^\star = \partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g})^t + \Delta t\ \ \partial_{\boldsymbol{x}} \cdot \left(-\partial_{\boldsymbol{x}} \cdot (\rho_g \boldsymbol{U_g} \otimes \boldsymbol{U_g}) + \partial_{\boldsymbol{x}} \cdot (\boldsymbol{\tau})\right). \tag{8.8}$$

  Therefore $\partial^2_{\boldsymbol{x}^2} \cdot (\tilde{P})$ can be computed.

- Velocity correction:

$$(\rho_g \boldsymbol{U_g})^{(t+\Delta t)} - (\rho_g \boldsymbol{U_g})^{\star} = -\Delta t \ \frac{1}{\gamma_\infty M^2} \partial_{\boldsymbol{x}} \cdot (\tilde{P}). \tag{8.9}$$

As far as numerical methods are concerned, the time resolution is provided by a third order explicit Runge Kutta scheme with minimal storage, (Wray 1990). Spatial evolution is done with a finite difference scheme, the derivatives are computed with a Pade 6^{th} order scheme, (Lele 1992). The boundary conditions are handled through the NSCBC method, (Poinsot and Lele 1992). The physical space solver is available for 2-D , Chpt. 10, 2-D axisymmetric, Chpt. 9, and 3-D configurations, Chpt. 11.

Spectral solver The Homogeneous Isotropic Turbulence (HIT) computations done in Chpt. 4 and Chpt. 11, are done with ASPHODELE's spectral solver. It solves an incompressible gas flow with a constant density, Eq. (1.106). The boundary conditions are periodic. The solver uses a deterministic forcing scheme conserving kinetic energy; its description as well as associated references are given in Sec. 4.1.1. This solver is used in this work to generate a steady velocity field. This velocity field is a good candidate in order to study multi-fluid computed spray dynamics, in 2-D and 3-D configurations, Chpt. 4 and Chpt. 11, respectively.

Lagrangian dispersed phase numerical scheme

The Lagrangian numerical particles ODE, Eq. (2.2), are solved within ASPHODELE with an explicit third order Runge Kutta solver. Therefore, the coupling between the gas and the liquid takes place at each Runge Kutta sub-step, in the following way:

- Lagrangian ODE resolution,
- Coupling term computations,
- Eulerian gas resolution.

The structure of one Runge Kutta sub-step is illustrated on Fig. 8.1, where we use the following definitions:

$$\begin{aligned} Drop^n &= \{(\boldsymbol{x}_k, \phi_k, \boldsymbol{u}_k, T_k)^n \, ; k = 1..N_p]\}, \\ S^n_{Drop} &= \{\left(S^m_{i,j,k}, S^{mom}_{i,j,k}, S^{enth}_{i,j,k}\right)^n ; i = 1..N_x,\ j = 1..N_y,\ k = 1..N_z\}, \\ Gas^n &= \{(\rho_{i,j,k}, \boldsymbol{U}_{i,j,k}, h_{i,j,k})^n \, ; i = 1..N_x,\ j = 1..N_y,\ k = 1..N_z\}, \end{aligned} \tag{8.10}$$

i.e., $Drop^n$ represents the numerical particles of the simulation at time t^n, S^n_{Drop} the Eulerian source terms of the gaseous equations computed from the droplets and discretized over the gaseous grid at time t^n. Finally, Gas^n represents the discretized gaseous fields at time t^n. The superscript $\star$ used in Fig. 8.1, indicates that the quantities are taken at time $t^\star$, an intermediate time of the Runge Kutta algorithm. We can see in the algorithm described in Fig. 8.1, that the gas is assumed constant within a Kunge Kutta sub-step. The main difficulties come from the coupling terms evaluation, that is tackled in the next paragraph.

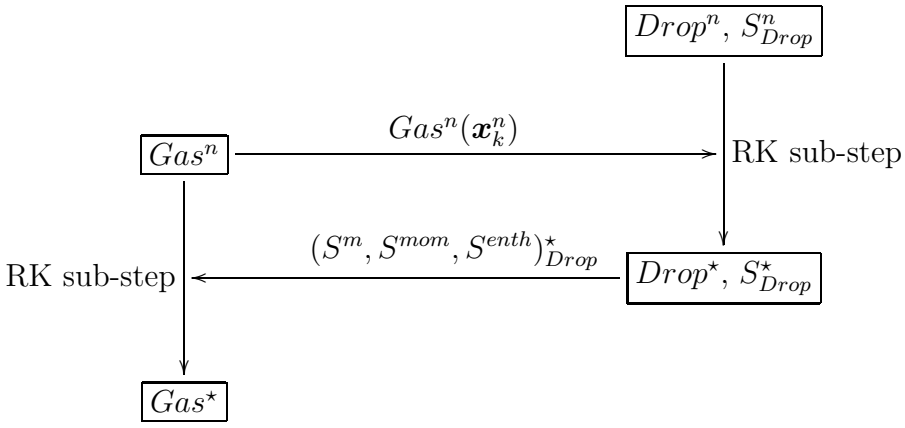


Figure 8.1 – *Runge Kutta sub-iteration for the coupled Eulerian-Lagrangian solver.*

8.1.2 Coupling Difficulties

Two main difficulties arise when coupling Eulerian and Lagrangian descriptions in two-phase flows. First, the gaseous quantities have to be computed at the position of the droplets in order to obtain the influence of the gas on the droplets: drag force, heat transfer, etc. Moreover, in the framework of dispersed liquid phase, the influence of the liquid on the gas is modeled by source terms obtained from the Lagrangian tracking of the droplets and thus needing to be projected on the Eulerian gaseous grid. These two drawbacks are linked to different issue. The first one leads to a significant cost increase, while the second one is related to an approximation issue. We present in this section, the way chosen to tackle these issues, as implemented in ASPHODELE, and thus used for the Lagrangian computations performed in Chpt. 9 to Chpt. 11.

Droplet location interpolation

The precision of the spray Lagrangian tracking, highly depends on the precision on the gas quantities seen by droplets. A third-order interpolation algorithm is then used in ASPHODELE, in order to compute the gaseous quantities at droplet location. This algorithm was first developed for Particle Image Velocimetry (PIV), (Lecordier, Demare, Vervisch, Reveillon, and Trinite 2001). This needed interpolation can be problematic in terms of computational cost in realistic configurations where millions of particles are tracked within a complex flow. Indeed, the interpolation has to be done at each Runge Kutta substep.

Lagrangian projection

Furthermore, a main difficulty of Eulerian/Lagrangian approaches is to project the Lagrangian quantities of droplets onto the Eulerian gaseous grid nodes. A PSI-CELL method, (Crowe, Sharma, and Stock 1977), is used in ASPHODELE to achieve this task. The Lagrangian contributions are instantaneously allocated to neighboring gas nodes, weighted by the distance to the nodes, see Fig. 8.2. This model may induces artificial diffusion, and the reasons of its use are discussed by (Péra 2005), and (Reveillon and Vervisch 2000), as well as possible improvements. A lot of work are devoted to this subject, see for instance

(Subramanian 2000).
A control volume $\mathcal{C}_\alpha$ is associated to each gas node and the Lagrangian contribution is obtained by weighted sum over the droplets affecting the considered node. For the mass source term, we obtain, see Sec. 2.1.3 for the details on droplet contribution:

$$S_j^m = \frac{\rho_l \, \mathrm{R_{drop\text{-}gas}}^{3/2}}{V_{\mathcal{C}_\alpha} \, 4\sqrt{\pi}} \sum_k \alpha_k^{(j)} \, S_k^{1/2} \, R_{S_k}, \tag{8.11}$$

where $\alpha_k^{(j)}$ is the distribution coefficient of the source term of droplet k on node j. This term, that can be seen as the portion of control volume of node j intersecting the control volume of droplet k, is given by, (Reveillon and Demoulin 2007):

$$\alpha_k^{(j)} = \frac{1}{V_{\mathcal{C}_\alpha}} \prod_{i=1}^{N_{\mathrm{dim}}} \left[\Delta_i^{(j)} - |x_i^{(j)} - x_{ki}| \right], \tag{8.12}$$

where $\Delta_i^{(j)}$ is the space step at the node j and in the i^{th} direction, and $x_i^{(j)}$ and x_{ki} the coordinates along the i^{th} direction of node j and droplet k respectively.

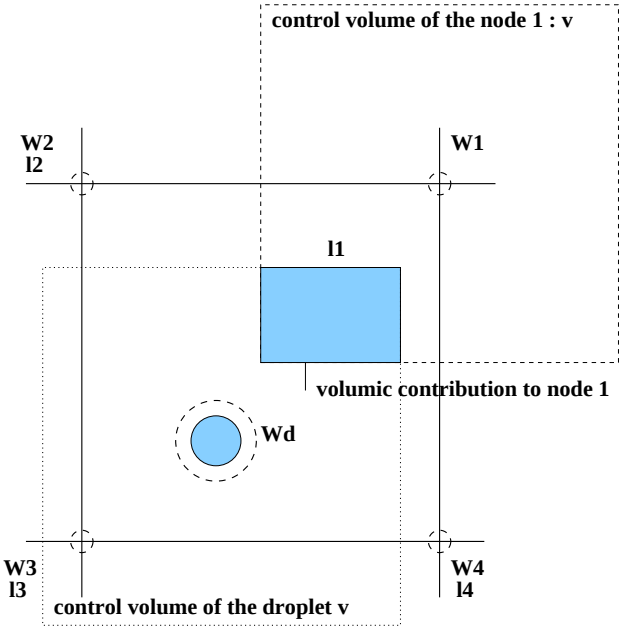


Figure 8.2 – *Two-dimensional sketch of the repartition of droplet source terms on the closest Eulerian nodes.*

8.2 Eulerian-Eulerian gas-liquid solver : ASPHODELE-MUSES3D coupling

The original academical solver used in this work, was obtained coupling ASPHODELE solver with the Eulerian multi-fluid solver MUSES3D developed in this work, and described Part. III. The first objective is to obtain a DNS gaseous solver to couple with MUSES3D, to assess multi-fluid model in more complex configurations, as HIT or jet configurations. The

second objective is to come up with a new numerical tool coupling Eulerian and Lagrangian descriptions of the spray with the same Eulerian DNS gas solver. We present here the coupling between the two Eulerian phases, before showing the general structure of the coupled code. This code allows precise comparisons between both description of the spray in a one-way coupling framework and, thanks to the FADO procedure, it allows to compare their evaporation in the gas phase, a key issue for combustion applications, and to perform Coldflame computations. This tool can naturally be extended to two-way configuration, albeit not allowing the same spray-description-focused comparisons, the gas phase being modified by the spray, but allowing to tackle combustion applications.

8.2.1 Euler-Euler solver one-way coupling

As far as time evolution is concerned, the coupling between both Eulerian methods differs from the one used for the Eulerian/Lagrangian coupling. Indeed, due to the kinetic scheme used for the multi-fluid, Chpt. 3, a simple "Euler-like" time step is enough to obtain the second order in time and space. Furthermore using a Runge Kutta scheme will not improve the global order of the method, due to the limitation of the space transport scheme order, see Chpt. 3. Finally, it would reduce the timestep and thus not keep a unity CFL for the physical transport. The constant gas assumption, valid for a Runge Kutta sub-step, can not be used for the whole time step. Therefore time interpolation of the gaseous fields are done to compute the Eulerian time evolution. It thus gives the structure presented in Fig. 8.3, where we used the definitions of Eq. (8.10), and further define:

$$MF^n = \left\{ \left(m^{(p)}_{i,j,k}, \boldsymbol{u}^{(p)}_{i,j,k}, h^{(p)}_{i,j,k} \right)^n ; p = 1..N_S,\ i = 1..N_x,\ j = 1..N_y,\ k = 1..N_z \right\}, \tag{8.13}$$

i.e., MF^n represents the multi-fluid Eulerian fields discretized over space and size and at time t^n The coupling presented in Fig. 8.3, is devoted to one-way coupling framework. We

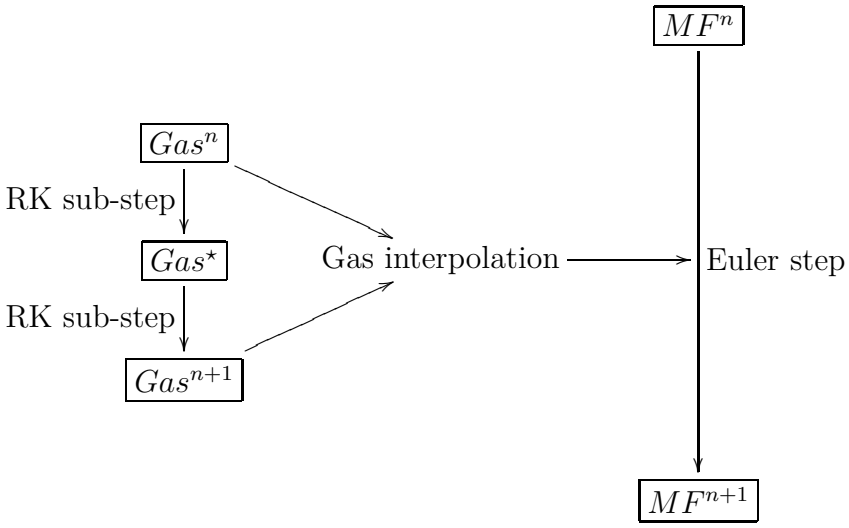


Figure 8.3 – *Time step for Eulerian (Gas)-Eulerian (multi-fluid spray) coupling. Gas phase evolves with Runge Kutta algorithm while multi-fluid phase with a single Euler step.*

discuss in the next paragraph the extension for the FADO procedure and for full two-way coupling.

8.2.2 Combining Euler-Euler and Euler-Lagrange approaches

We present here the global structure of the numerical solver, first in a one-way framework. In this context, the same gas phase can be used for both Eulerian and Lagrangian descriptions, allowing precise evaluation focusing on the spray methods, and to conduct computational efficiency studies, realizing Eulerian and Lagrangian spray computations in the same code run. The coupling structure is slightly modified in the framework of the FADO procedure. The gas phase is still not modified, but, through the use of scalars, we can assess the description of fuel vapor formation in each spray method. These scalars can also react in Coldflame computations, as described in Sec. 1.4.3, and used in Chpt. 10.

One way framework

The standard one way framework is obtained plugging Eulerian and Lagrangian spray description on to the Eulerian gas description, and is described by Fig. 8.4, using the definitions introduced in Fig. 8.1 and Fig. 8.3. This structure is used in Chpt. 9 and Chpt. 11,

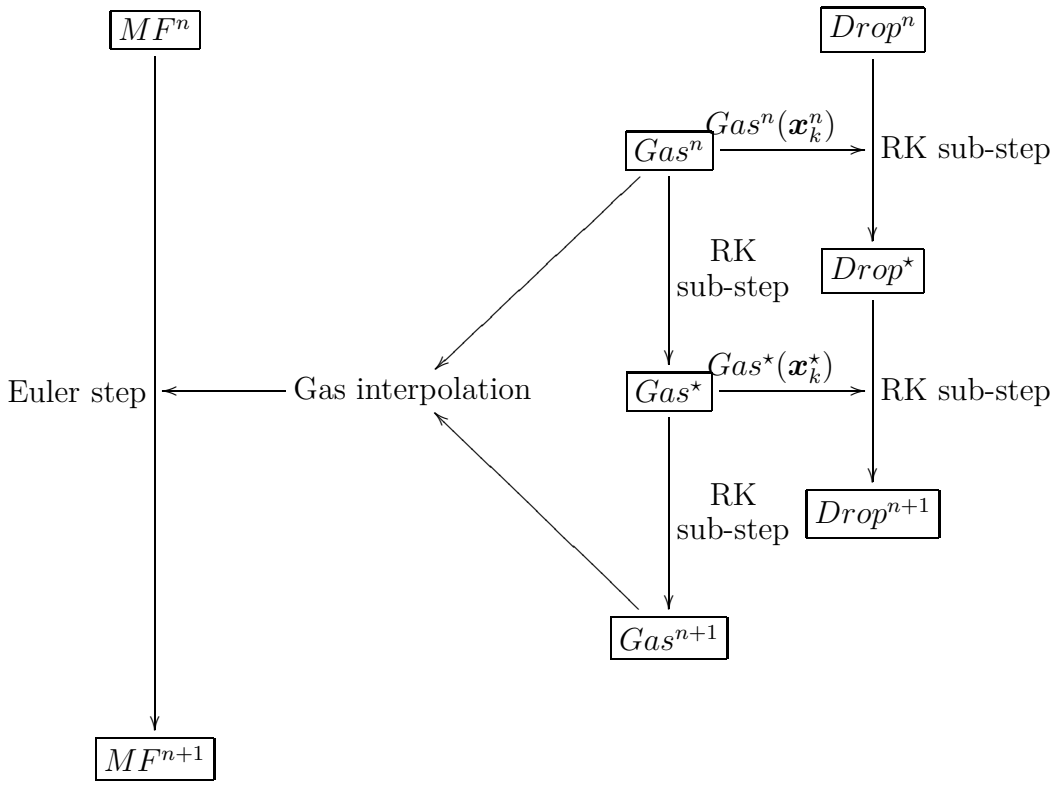


Figure 8.4 – *Time step for Euler/Euler-Euler/Lagrange solver in one-way framework. Gas phase is computed only once for both spray description.*

to evaluate the ability of the multi-fluid model to reproduce the size-conditioned dynamics of a spray in 2-D axi-symmetrical and 3-D configurations. The issue of computational cost is raised in Chpt. 9.

Gaseous fuel vapor prediction tools design

When using FADO procedure, we solve the evolution of the fuel mass fraction, obtained by the spray descriptions. The coupling between ASPHODELE and MUSES3D, allows to compute the fuel mass fraction obtained from the Lagrangian, $Y_{F'}^{Drop}$ and from the Eulerian multi-fluid method, $Y_{F'}^{MF}$. We recall here the equation of the fuel mass fraction $Y_{F'}$ evolution, Sec. 1.4.3:

$$\partial_t \left(\rho_g Y_{F'}\right) + \partial_{\boldsymbol{x}} \cdot \left(\rho_g \left(\boldsymbol{U_g} + \boldsymbol{U_g}^{\text{cor}}\right) Y_{F'}\right) = \partial_{\boldsymbol{x}} \cdot \left(\rho_g D^{\star}_{Y_{F'}} \partial_{\boldsymbol{x}} \cdot \left(Y_{F'}\right)\right) + \dot{\omega}_{Y_{F'}} + S^m. \tag{8.14}$$

The scalar is advected and diffused by the flow, fed by the spray evaporation and consumed in the case of the Coldflame computation. In the case of the Lagrangian spray description, the time evolution of $Y_{F'}^{Drop}$ is resolved through a Runge Kutta algorithm, and is thus easily introduced within the coupling algorithm, Fig. 8.1, as shown in Fig. 8.5. In the case of the

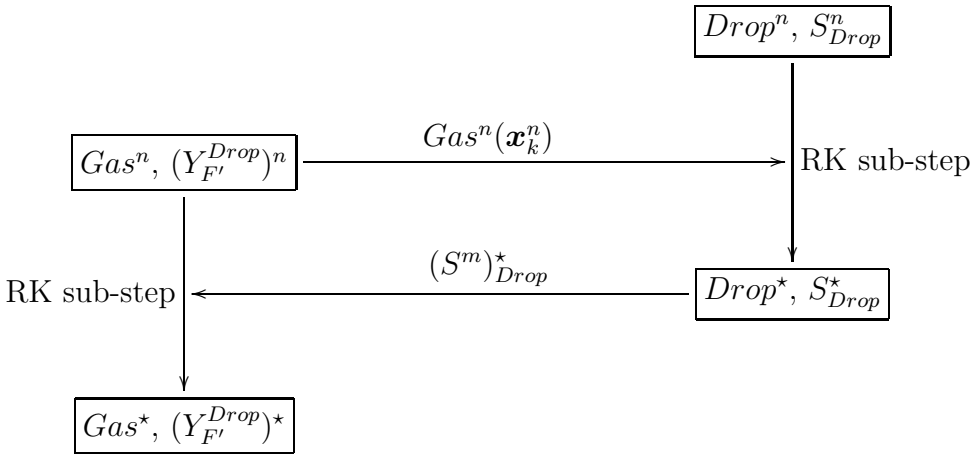


Figure 8.5 – *Runge Kutta sub-iteration for the coupled Eulerian-Lagrangian solver, with fuel mass fraction computation, using the FADO procedure.*

Eulerian multi-fluid spray description, the insertion of the fuel mass fraction in the coupling algorithm is not so natural. Indeed, we want to use the same Runge Kutta algorithm for the evolution of the scalar, nevertheless, the spray being only resolved within a Euler time step, we can only compute the mass source term, S^m, at time t^{n+1}, and not at the intermediate times of the Runge Kutta algorithm. We therefore decide to resolve the evolution of the fuel mass fraction using an operator splitting: solving first the evolution due to transport, diffusion and reaction:

$$\partial_t \left(\rho_g Y_{F'}\right) + \partial_{\boldsymbol{x}} \cdot \left(\rho_g \left(\boldsymbol{U_g} + \boldsymbol{U_g}^{\text{cor}}\right) Y_{F'}\right) = \partial_{\boldsymbol{x}} \cdot \left(\rho_g D^{\star}_{Y_{F'}} \partial_{\boldsymbol{x}} \cdot \left(Y_{F'}\right)\right) + \dot{\omega}_{Y_{F'}}, \tag{8.15}$$

through a Runge Kutta algorithm. The evolution due to spray evaporation is solved in a second time:

$$\partial_t \left(\rho_g Y_{F'}\right) = S^m, \tag{8.16}$$

through a one-step Euler method. We choose a Lie splitting algorithm, leading to:

- solve Eq. (8.15) during Δt, to obtain $(\widetilde{Y_{F'}^{MF}})^{n+1}$,
- solve Eq. (8.16) during Δt, to obtain $(Y_{F'}^{MF})^{n+1}$.

The resulting coupling algorithm is presented in Fig. 8.6. The complete algorithm, for both types of coupling, in the context of the FADO procedure, is summarized in Fig. 8.7.

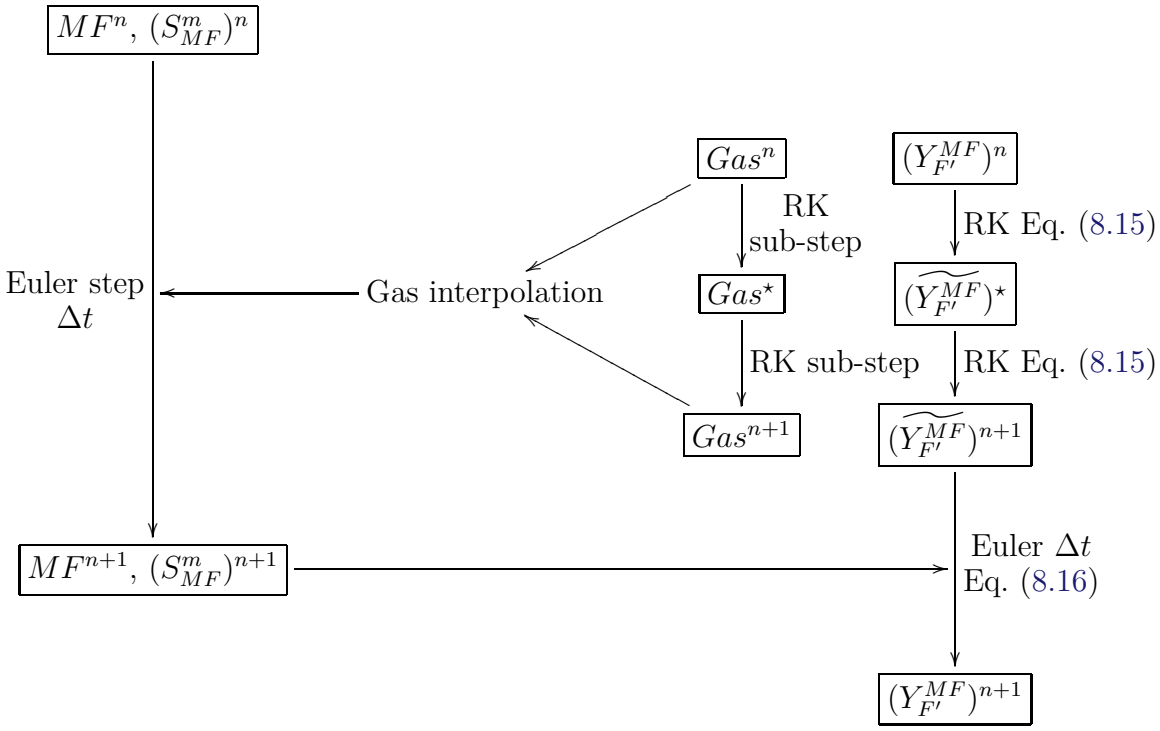


Figure 8.6 – *Time step for the coupled Eulerian-Eulerian solver, with fuel mass fraction computation, using the FADO procedure.*

Full two-way coupling

One has to note that, albeit algorithms presented in this study are devoted to one-way coupling, there is no limitation to extend to two-way framework. A first way would be to use a Runge Kutta scheme for the multi-fluid evolution and to couple for each Runge Kutta sub-step, as done for the Lagrangian coupling, Fig. 8.1. Nevertheless it induces an additional cost for the multi-fluid method, without any precision improvement. Furthermore, the gas has to be assumed constant in a sub-step. Another way to realize a such coupling would be to generalize the operator splitting used for the multi-fluid, Sec. 3.2, to the gas phase resolution. A global splitting algorithm could then be proposed, separating for the gas phase resolution the fuel source terms evaluation from the rest of the system:

- solve transport, diffusion and reaction for the gas and transport for the spray during $\Delta t/2$,
- solve phase space transport for the spray, and evolve gas phase due to fuel source terms during Δt
- solve transport, diffusion and reaction for the gas and transport for the spray during $\Delta t/2$.

The recently provided operator splitting techniques for low Mach number flames with complex chemistry, (Descombes, Dumont, Louvet, Massot, Laurent, and Beaulaurier 2009), could be used for this case of splitting between low Mach flow and spray resolution.

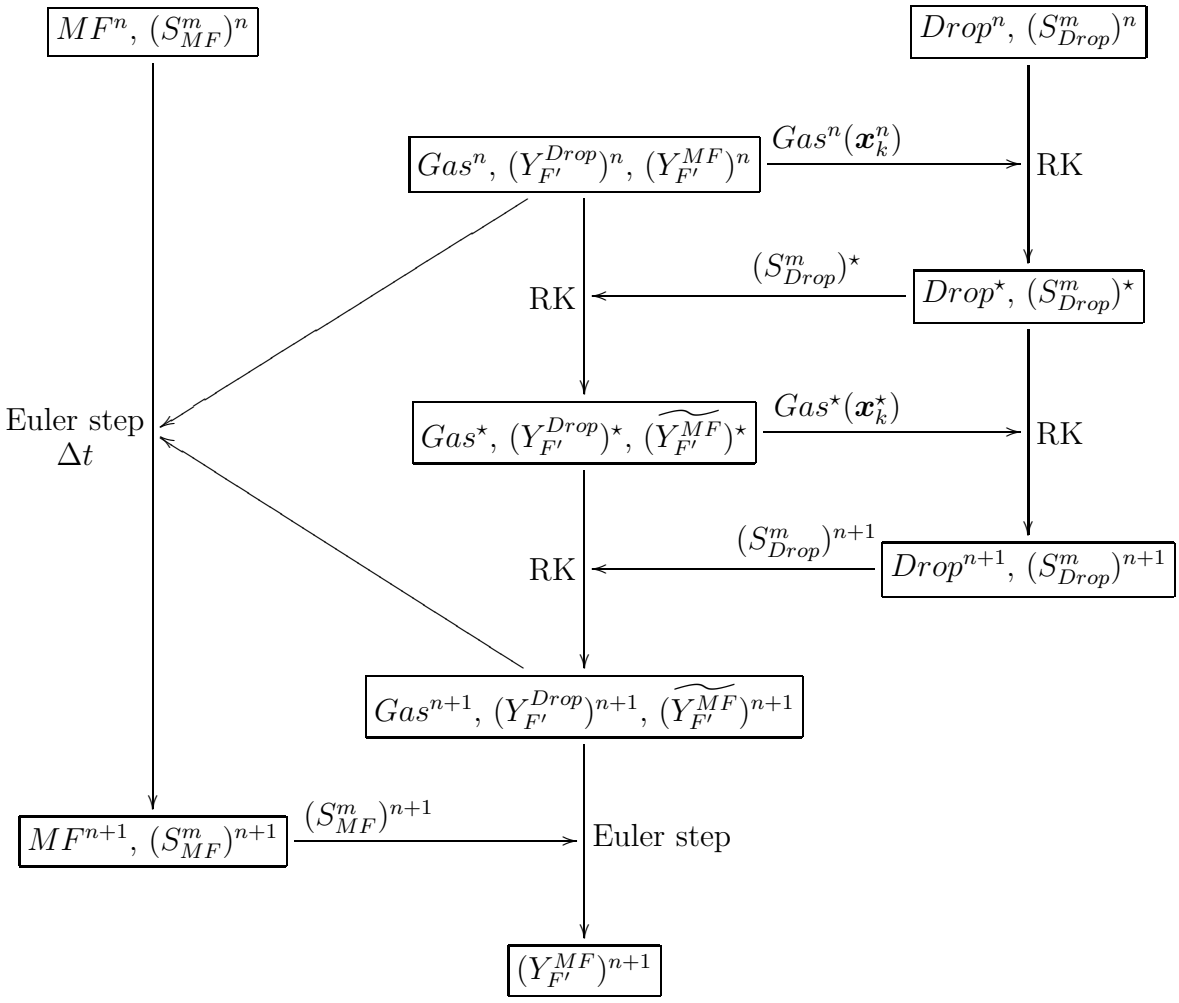


Figure 8.7 – *Time step for Euler/Euler-Euler/Lagrange solver with the FADO procedure. The gas phase is computed only once for both spray description.*

Conclusion

A new academical solver devoted to spray resolution evaluation is provided, coupling ASPHODELE and MUSES3D solvers. It combines Eulerian/Eulerian and Eulerian/Lagrangian approaches, as well as a framework allowing to isolate spray resolution effects. This framework allows to tackle spray combustion applications. This solver is used in Chpt. 9 to compare Eulerian multi-fluid and Lagrangian spray descriptions. Precision considerations, for the size-conditioned dynamics prediction, as well as computational cost issues are tackled. The precision evaluation is complemented in Chpt. 10, where the essential issue of fuel vapor mass fraction prediction is assessed. Finally, the feasibility of 3-D computations with the provided solver, is assessed in Chpt. 11.

Chapter 9

Eulerian multi-fluid versus Lagrangian for two-phase jet simulation

Contents

Given the industrial concerns driving this study, jet configurations have to be tackled. Furthermore, a 2-D axisymmetric configuration includes some three dimensional effects and is thus a first step toward industrial spray injection configurations. In this chapter, the multi-fluid method is evaluated in a such configuration. The computations conducted use the coupling between ASPHODELE and MUSES3D solvers, Chpt. 8. This chapter aims at showing the ability of the multi-fluid method to describe the global dynamics of a polydisperse spray injected by the gaseous jet. Two main issues are evaluated for this purpose. First, the ability of the multi-fluid method to describe the polydispersion of the spray, in non-evaporating and evaporating cases, is illustrated. This essential point will be complemented in Chpt. 10, where the key issue of the fuel vapor mass fraction description is tackled. Second, the global behavior of the method in the case where the monokinetic assumption is locally broken is studied. In order to realize this evaluation, a Lagrangian method is chosen as a reference. Comparisons between the Eulerian and Lagrangian spray size-conditioned dynamics are conducted. Finally, the computational cost associated to the Eulerian multi-fluid method is studied, in order to argue on the feasibility of this method. This study was conducted within the TIMECOP-AE European project led by SAFRAN Turbomeca, (de Chaisemartin, Laurent, Massot, and Reveillon 2008a).
The Eulerian/Lagrangian comparisons conducted in this chapter bring in two main novelties. First the comparisons are time-resolved. Indeed, instantaneous fields, and not averaged, are compared. Second, the comparisons are provided for the size-conditioned dynamics, a key issue for combustion applications, Chpt. 10

9.1 Axisymmetric free jet with polydisperse spray injection

The studied configuration is designed to enable the evaluation of the Eulerian multi-fluid method on an unsteady jet. Turbulence is injected to destabilize the gasesous jet. Furthermore a polydisperse spray is injected to study the multi-fluid size distribution description. The Lagrangian reference is not taken as a converged Direct Simulation Monte-Carlo method in this study, in order to be closer to industrial concern. Given the DNS configuration used, we perform a Discrete Particle Simulation. Indeed there is no need to use Stochastic Parcel method, since all the droplets contained in the computational domain can be tracked. In order to focus on the Eulerian method evaluation, a one-way coupling is considered, and simple droplet models, are used.

9.1.1 Gaseous jet

A jet was chosen in order to be closer to the aimed industrial applications, i.e., fuel spray injection in combustion chambers. The axisymmetric metric allows to account for some 3-D effects.

The gas jet is computed with the low Mach dilatable solver available in ASPHODELE, Chpt. 8, solves the system Eq. (1.105), with the numerical methods presented in Chpt. 8. The jet is destabilized with turbulence injection through a Klein's method with 5% of fluctuations. We define the Reynolds number based on the geometry:

$$\mathrm{Re}_0 = \frac{U_0\, x_0}{\nu_\infty}, \tag{9.1}$$

where U_0 is the injection velocity, and x_0 is the jet width. We have $\mathrm{Re}_0 = 1000$ in the presented simulations, that corresponds to a jet width $x_0 = 1.5$cm, an injection velocity $U_0 = 1$m/s for a typical kinematic viscosity $\nu_\infty = 1.6 \times 10^{-5}\mathrm{m}^2/\mathrm{s}$. The complete domain of the simulation has the dimension $12x_0 \times 6x_0$, but due to the symmetry, we only compute half of the radial direction: $12x_0 \times 3x_0$, with 400×100 grid points. The results are presented at dimensionless time $t = 15$, i.e. the jet has already gone across the computational domain once. The gas vorticity is presented in Fig. 9.1.

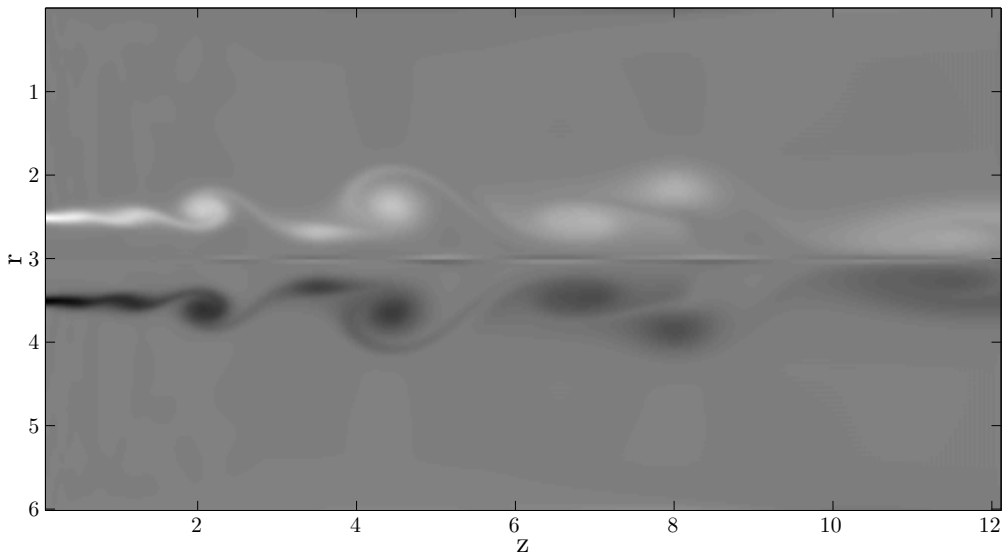


Figure 9.1 – *Gaseous vorticity at dimensionless time t=15, obtained with a low mach number dilatable solver on a* 400×100 *grid.*

9.1.2 Eulerian polydisperse spray injection

The spray injected in the gaseous jet is polydisperse to study the multi-fluid size distribution resolution. We choose a log-normal distribution to represent its polydispersion, Fig. 9.2. Evaporating and non-evaporating configurations are raised, to study the polydispersion resolution of the multi-fluid with and without coupling terms between the sections. The

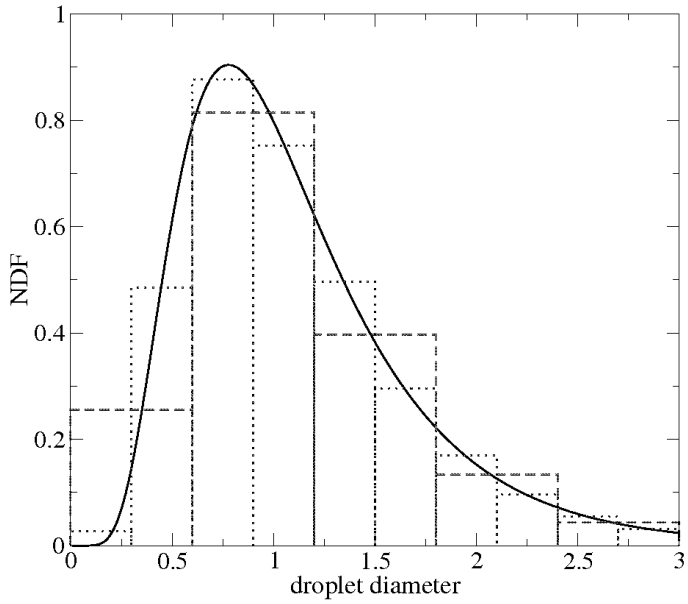


Figure 9.2 – *Spray injection in free jet configuration: polydisperse log-normal distribution discretized with 5, dashed lines, to 10, dotted lines, sections.*

spray is injected at the gas injection velocity. As far as droplet models are concerned, Chpt. 1, we still choose to work in a framework of simple models as well as one-way coupling, in order to focus on the resolution methods. Thus, we take a D^2-law associated with a constant Spalding number $B_M = 0.1$, Sec. 1.2.2. Besides, we only consider a Stokes drag force as external forces. Finally, droplet interactions are not accounted for.

Among the different multi-fluid versions, we use a "classical" monokinetic multi-fluid method with a first order evaporation scheme, Sec. 2.3, since it is the multi-fluid method with the highest maturity and relative simplicity, and therefore the more likely to be implemented in semi-industrial or industrial codes. The specific kinetic scheme ensuring the needed robustness for the multi-fluid is used in its axisymmetric form developed in this study, Sec. 3.2. For the coupling with the gas phase, the same grid is chosen for the multi-fluid, so that no spatial interpolation is needed. As far as size phase space is concerned, we use five sections for the non-evaporating case and ten for the evaporating one, Fig. 9.2.

9.1.3 Reference method and Eulerian multi-fluid evaluation

A DPS Lagrangian method, Chpt. 2, is chosen to evaluate the efficiency of the multi-fluid. In this framework, a numerical particle represents one physical droplet. The number of particles is determined by the stoichiometry. At the considered time, we have 10000 up to 40000 droplets in the computational domain.

We are interested in the global behavior of the Eulerian spray description and we will thus conduct in this part qualitative comparisons between both approaches. We will compare the droplet positions from the Lagrangian DPS computation to the Eulerian number density computed with the multi-fluid method. The DPS can be seen in this case as one realization of the DSMC method, i.e., one realization of an ensemble average driven by the Williams equation, Chpt. 2. This point of view is adopted to justify comparisons with the multi-fluid

model derived from a statistical spray description driven by the Williams equation, Chpt. 1 and Chpt. 2. Thanks to the multi-fluid polydispersion resolution, we will perform the comparisons for different ranges of sizes and thus for different inertia, in the evaporating and non-evaporating cases. Finally, we discuss the computational efficiency issue. We compare the cost of a DPS to the cost of a multi-fluid computation. Moreover, we present the cost of more refined DMSC computations, in order to have a Lagrangian description at the same level as the multi-fluid one. Indeed, the Eulerian statistical fields can not be recovered from the DPS computation, as the ensemble averages can not be obtained directly, whereas the DSMC gives access to them. One can note that in the infinite Knudsen limit, i.e., no droplet interaction, the DSMC computation is equivalent to an ensemble of DPS, each numerical particle representing one droplet and having a weight equal to one.

9.2 Lagrangian vs. multi-fluid for evaporating and non-evaporating spray

9.2.1 Non-evaporating spray injection

For this test case we used five sections for the Eulerian multi-fluid simulation. We have 40000 Lagrangian particles in the computational domain. We first present the results for the whole spray, regardless of the droplet sizes. These results are plotted on Fig. 9.3, where the Lagrangian particle positions are plotted over the gas vorticity field and compared to the Eulerian total number density. The global droplets dynamics are very well reproduced

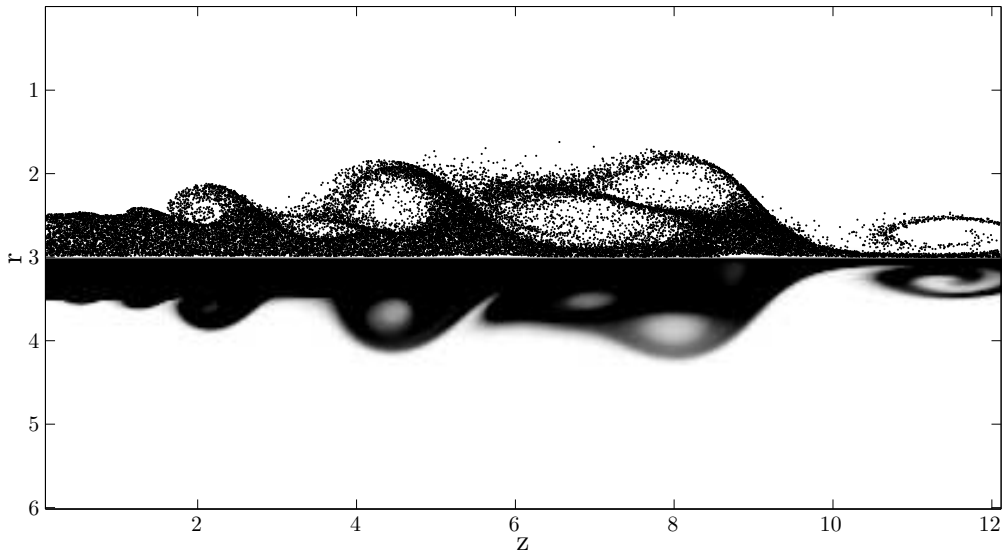


Figure 9.3 – ***Non evaporating polydisperse spray****, Stokes* 0.003 *to* 4.45 *corresponding to diameters* $d = 4\,\mu m$ *to* $d = 150\,\mu m$*, at dimensionless time t=15, : (top) Lagrangian particle positions with* 40000 *particles over the gas vorticity, (bottom) Eulerian total number density with a* $400 \times 100 \times 5$ *grid.*

by the Eulerian multi-fluid method. We move then to size-conditioned dynamics. We

found a very good agreement for the droplets with low inertia, for Stokes from 0.003 to 0.18 corresponding to diameters from $4\mu m$ to $30\mu m$, as shown in Fig. 9.4. This shows the ability of the multi-fluid method to simulate the dynamics of a spray where few crossings occur, the droplet dynamics being close to the gas one, therefore in its validity domain, Sec. 2.3. For

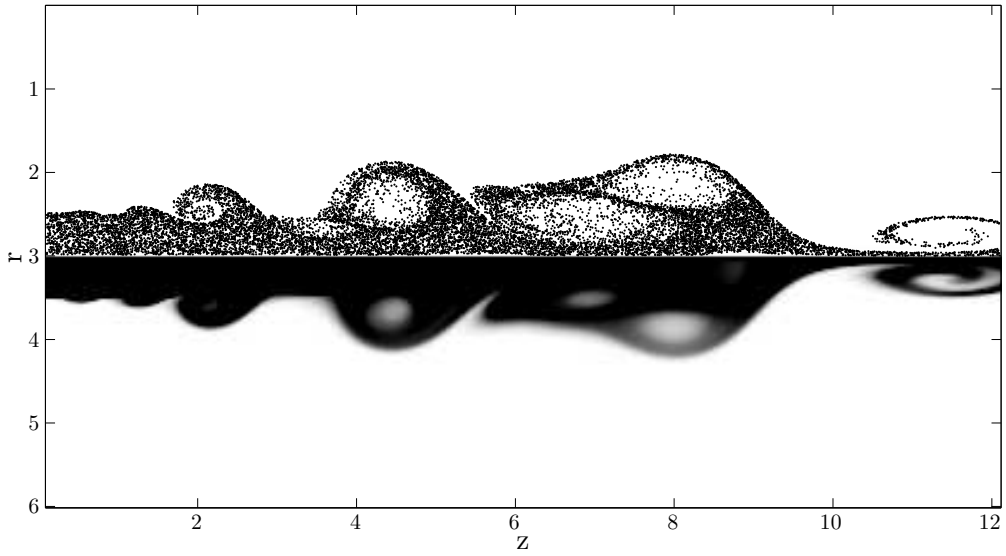


Figure 9.4 – ***Non evaporating polydisperse spray, low inertia droplets****, Stokes* 0.003 *to* 0.18 *corresponding to diameters* $d = 4\,\mu m$ *to* $d = 30\,\mu m$*, at dimensionless time t=15 : (top) Lagrangian particle positions with 40000 particles over gas vorticity, (bottom) Eulerian number density with a* $400 \times 100 \times 5$ *grid.*

higher Stokes number, the droplets are ejected from the vortices and crossing trajectories are likely to occur, breaking the monokinetic multi-fluid assumption. Nevertheless, the dynamics is still very well reproduced for higher Stokes number. The results are plotted in Fig. 9.5 for Stokes number from 0.71 to 4.45 corresponding to diameters from $60\,\mu m$ to $150\mu m$. One can notice that the number density is concentrated in a few cells in this case and the numerical method does not encounter any problem to capture it, illustrating again its robustness.

9.2.2 Evaporating spray injection

This weakly turbulent free jet is also assessed with an evaporating spray. The results are presented the way chosen for the non-evaporating case. In order to describe correctly the evaporation process, we took ten sections for the Eulerian multi-fluid simulation. In this case, 10000 Lagrangian particles are present in the domain at the considered time. As in the non-evaporating case, we found a very good agreement between the Eulerian multi-fluid and the Lagrangian descriptions, see Fig. 9.6. The size-conditioned dynamics are still very well predicted by the multi-fluid method as well for low Stokes number shown in Fig. 9.7, as for high Stokes number shown in Fig. 9.8. These comparisons show the ability to treat the size distribution evolution through the multi-fluid coupled sections.

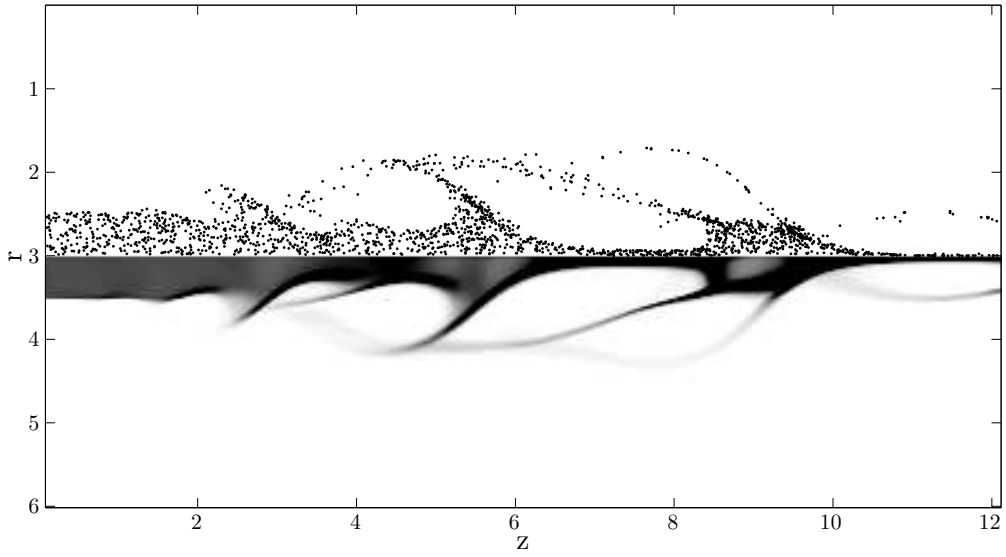


Figure 9.5 – ***Non evaporating polydisperse spray, high inertia droplets****, Stokes* 0.71 *to* 4.45 *corresponding to diameters* $d = 60\,\mu m$ *to* $d = 150\,\mu m$*, at dimensionless time t=15 : (top) Lagrangian particle positions with 40000 particles over gas vorticity, (bottom) Eulerian number density with a* $400 \times 100 \times 5$ *grid.*

This polydisperse evaporating axisymmetric free jet shows the ability of the multi-fluid method to treat more complex case, closer to industrial configuration. These first comparisons with Lagrangian computations show the ability of the Eulerian method to capture the size-conditioned dynamics in this case and are very encouraging. Further results concerning polydisperse spray Eulerian/Lagrangian comparisons, for example regarding gaseous fuel mass fraction, are presented in Chpt. 10, as well as Coldflame, computations.

9.2.3 Eulerian vs. Lagrangian: computational efficiency

In order to give a relevant evaluation of the multi-fluid cost in this jet configuration, we compare it to the cost of a Lagrangian DPS. Moreover, in order to complement this study, we also analyze the cost of more refined DSMC Lagrangian computation, since they can achieve, with enough numerical particles, the same level of description as the Eulerian multi-fluid method. We thus define different levels of resolution for both methods. As far as the Lagrangian method is concerned, we defined three different types of computations. We consider first a Discrete Particle Simulation with 40000 droplets at the end of the computation. As mentioned earlier, we consider this computation as one realization, driven by the stoichiometry, of the statistical DSMC Lagrangian method. Even if this level of modeling does not correspond to the one provided by the multi-fluid method, DPS provides, in DNS configuration, a cost closer to the one of an industrial Stochastic Parcel method. It is thus important to compare DPS and Eulerian multi-fluid costs. Second we study the cost of statistical DSMC Lagrangian computations with 1 Million and 4 Million numerical particles. We defined the statistical computation with 4 Million particles as the refined Lagrangian computation. This refinement allows a limited noise on an Eulerian projection,

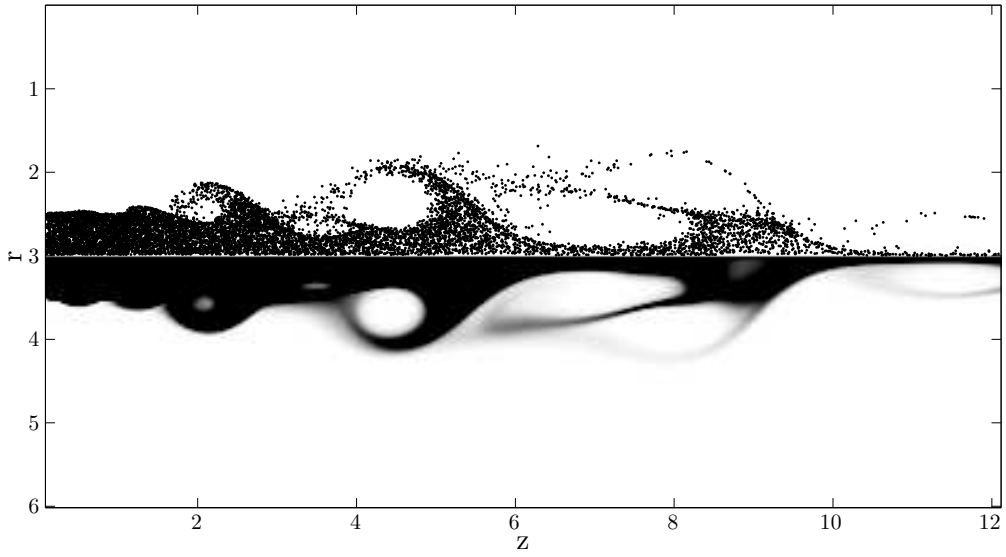


Figure 9.6 – ***Evaporating polydisperse spray****, Stokes* 0.003 *to* 4.45 *corresponding to diameters* $d = 4\,\mu m$ *to* $d = 150\,\mu m$*, at dimensionless time t=15, : (top) Lagrangian particle positions with* 40000 *particles, (bottom) Eulerian total number density with a* $400 \times 100 \times 10$ *grid.*

as seen in the Isotropic Homogeneous Turbulence case in Fig. 4.21, where we projected the Lagrangian results on the Eulerian grid. We compared this refined Lagrangian computation to an Eulerian multi-fluid one computed on a refined 800×200 grid with 20 sections. The DPS and the intermediate statistical Lagrangian computations are compared with the multi-fluid on a 400×100 grid with 10 sections. The results are summarized in Table 9.1.

As far as statistical Lagrangian description is concerned, the Eulerian multi-fluid method presents the same cost in the intermediate case and is twice more costly in the refined case. Nevertheless one has to notice that the $400 \times 100 \times 10$ grid already produced refined results, as one can see on Fig. 9.8 where very thin spray number density structures are captured. Furthermore, as shown in Sec. 4.3, from a statistical point of view, the 4 Million particles Lagrangian computation still produced, even if limited, some noise. This can be seen on Fig. 4.21 for the isotropic homogeneous turbulence case, and this statement stands for the jet case. Therefore more numerical particles would be needed to obtain better statistics and the multi-fluid is thus very competitive compared to the Lagrangian DSMC method. As far as Discrete Particle Simulation is concerned, the Eulerian multi-fluid presents higher cost, as seen in Table 9.1. Nevertheless one has to notice the high parallelization capability of Eulerian methods. The multi-fluid model will allow us to parallelize efficiently the computation, as described in Chpt. 7 and used in a 3-D case in Chpt. 11. It increases very significantly the efficiency of the Eulerian computations, and makes it very competitive even compared to a sequential Lagrangian Discrete Particle Simulations.

In the present work, the two methods are only compared for the same one-way configuration and for a given velocity field. Therefore, the computational cost purely devoted to the two-way coupling of the phases is not evaluated. It is clear that such a topic would be necessary, and we can envision that such a coupling will be very advantageous for the Eulerian method since the exchange of mass, momentum and heat for the Lagrangian approach have to

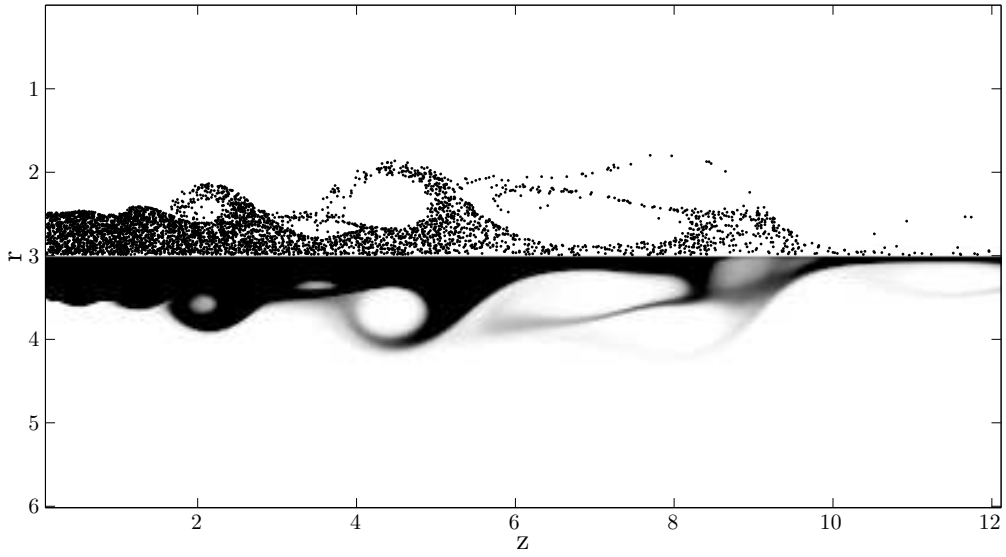


Figure 9.7 – ***Evaporating polydisperse spray, low inertia droplets****, Stokes* 0.003 *to* 0.18 *corresponding to diameters* $d = 4\,\mu m$ *to* $d = 30\,\mu m$*, at dimensionless time t=15 : (top) Lagrangian particle positions with 40000 particles, (bottom) Eulerian number density with a* $400 \times 100 \times 10$ *grid.*

be distributed to the various Eulerian nodes of the gaseous phase, due to heterogeneous descriptions. Besides, such a problem also results in a kind of numerical "diffusion" related to the Lagrangian approach, Chpt. 8. Even if speculative, such a statement allows to predict that in such a context, for two-way coupling, the Eulerian description will be cheaper in terms of computational cost without leading to any level of oscillation once the mesh is coarsened. Such a statement is consistent with the conclusions from (Druzhinin and Elghobashi 1999).

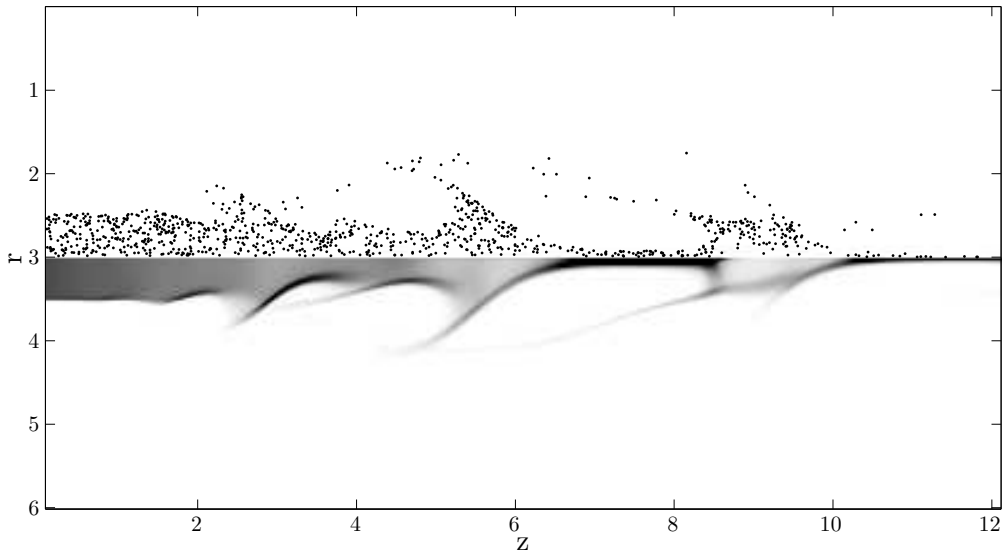


Figure 9.8 – ***Evaporating polydisperse spray, high inertia droplets***, *Stokes* 0.71 *to* 4.45 *corresponding to diameters* $d = 60\,\mu m$ *to* $d = 150\,\mu m$*, at dimensionless time t=15 : (top) Lagrangian particle positions with 40000 particles, (bottom) Eulerian number density with a* $400 \times 100 \times 10$ *grid.*

Lagrangian precision	DPS Level 40 000 parcels	Intermediate 1 Million parcels	Refined Sim. 4 Million parcels
Lagrangian CPU cost ($\times 1000s$)	4	20	80
Eulerian precision	400x100x10	400x100x10	800x200x20
Eulerian CPU cost ($\times 1000s$)	20	20	160
Ratio Eulerian over Lagrangian	**5**	**1**	**2**

Table 9.1 – *Computational cost comparisons between the Lagrangian and Eulerian multi-fluid methods. 2D axisymmetric free jet with polydisperse evaporating spray injection.*

Chapter 10

Toward combustion with Eulerian multi-fluid spray description

Contents

The applications aimed at with the multi-fluid method are combustion applications, as aeronautical or automotive combustion chambers where liquid fuel is burnt. In this work, the attention is focused on spray resolution methods, and thus the spray resolution has been isolated from the gas phase one, for example taking one-way description. Nevertheless, we would like to provide a first step toward combustion applications with the multi-fluid method. A key point for two-phase combustion, when considering homogeneous regime, is to compute precisely the fuel vapor mass fraction. This chapter evaluates the relevance of the multi-fluid method for combustion applications. A Cartesian free jet configuration is used, where the Eulerian description is first validated using the procedure introduced in Chpt. 9. The multi-fluid description of the fuel vapor mass fraction is then evaluated, still focusing on the spray resolution method. A procedure to simulate the fuel vapor mass fraction without modifying the gas phase is thus used, storing the evaporated fuel in a scalar advected and diffused by the flow. This procedure is described as the FADO procedure in Chpt. 1. It allows to conduct precise comparisons with a Lagrangian description of the fuel vapor mass fraction, concentrating on spray resolution issues. Finally, although this method does not allow to conduct standard combustion computation, a Coldflame, neglecting the flame effect on the gas phase, can yet be simulated, leading to a first multi-dimensional two-phase combustion with a multi-fluid spray description.
This multi-fluid evaluation devoted to combustion applications has been conducted during the 2008 CTR Summer Program at Stanford, (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009).
This study brings in original Eulerian/Lagrangian comparisons for combustion applications and provides a new spray method evaluation framework devoted to combustion concerns.

10.1 Cartesian free jet: Eulerian description validation

We decide to conduct these study in the context of a 2-D plane jet with spray injection. It provides a configuration easy to set up and well thus suited for first validations of fuel mass fraction computations. It is a less rich configuration than the 2-D axisymmetric jet, nevertheless it still contains interesting physics within an easier context. Before studying the fuel vapor coming from the injected spray evaporation, we first validate the multi-fluid method in this jet, performing Lagrangian comparisons, as introduced in Chpt. 9.

10.1.1 Cartesian free gaseous jet with spray injection

The low Mach dilatable solver implemented in CORIA's ASPHODELE, Chpt. 8, is here used to compute the plane jet. Turbulence is injected through a Klein method with 10% of

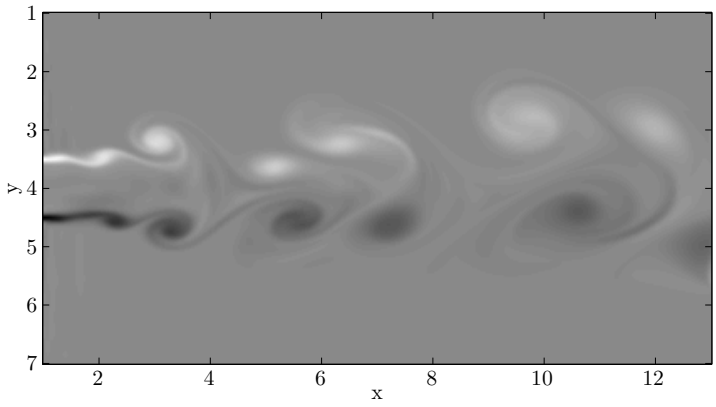


Figure 10.1 – *Free jet configuration: gaseous vorticity at time t=20, obtained on a* 400×200 *grid.*

fluctuations. The Reynolds number based on the geometry, as defined for the axisymmetric jet Chpt. 9, is 1000. The computational domain has the size $12x_0 \times 6x_0$, where $x_0 = 1.5\text{m}^{-2}$ is the injection width. The computational grid has 400×200 points. The vorticity field is plotted in Fig. 10.1 at dimensionless time $t = t\,U_0/x_0 = 20$, where, as in the axisymmetrical case, $U_0 = 1\text{m/s}$ is the jet injection velocity. As far as the liquid spray is concerned, N-heptane is injected in the jet core, with a log-normal distribution, Fig. 9.2. We use the modeling framework described for the axisymmetric jet, Chpt. 9, that is a D^2-law for evaporation, with a constant Spalding number $\text{B}_\text{M} = 0.1$, and a Stokes drag force, neglecting other forces, droplet interactions and computing a one-way coupling.

10.1.2 Eulerian plane jet computation validation

Before studying the fuel vapor issue, the Eulerian description of the spray dynamics is validated, using Lagrangian qualitative comparisons, as done in the case of the axisymmetric jet, Chpt. 9. With regard to the multi-fluid method, the "classical" monokinetic multi-fluid method is used, with five to ten sections for the size phase space discretization, for non-evaporating and evaporating cases, respectively. To facilitate coupling with the gas phase, the same grid is used for the gas and for the multi-fluid method, with 400×200 points. The splitting-based specific numerical method used in this 2-D Cartesian case is described in details in Chpt. 3. As far as the Lagrangian reference is concerned, a DPS is chosen, see Chpt. 9 for motivation, with 10000 to 70000 droplets at the considered time.

The global behavior of the spray in the non-evaporating and evaporating cases is well predicted by the multi-fluid method, as shown in Fig. 10.2 and Fig. 10.3. To assess the behavior of the multi-fluid, one has to check its ability to describe size-conditioned dynamics. Low inertia droplet dynamics are plotted in Fig. 10.4 for both the Eulerian and Lagrangian descriptions in the non-evaporating case. The Stokes range goes from $\text{St} = 0.011$ to $\text{St} = 0.12$, corresponding to diameters from $D = 9\mu\text{m}$ to $D = 30\mu\text{m}$. Very good qualitative agreement is still to be found for these dynamics descriptions. Furthermore, as done for the axisymmetric jet, one has to validate the dynamics of higher inertial droplets. The dynamics of droplets with Stokes from $\text{St} = 0.48$ to $\text{St} = 1.1$, i.e., diameters from $D = 60\mu\text{m}$ to $D = 90\mu\text{m}$, is therefore studied. In the present case, the turbulence

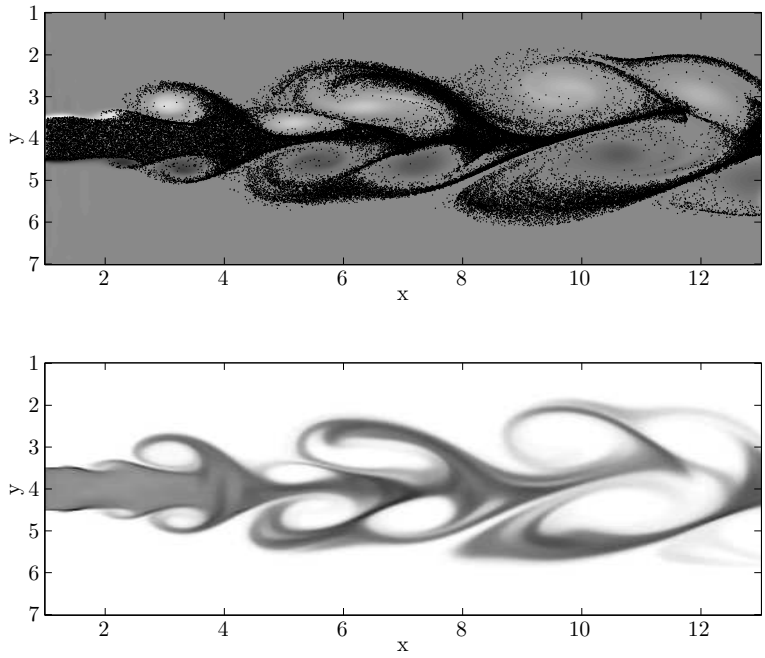


Figure 10.2 – ***Non-evaporating polydisperse spray****, whole size distribution, Stokes* 0.011 *to* 1.1 *corresponding to diameters* $D = 9\mu m$ *to* $d = 90\mu m$*, at time t=20: (top) Lagrangian particle positions with 70000 particles, (bottom) Eulerian multi-fluid monokinetic number density on a* $400 \times 200 \times 5$ *grid.*

level being slightly higher, droplet trajectory crossings are more likely to occur. A very good precision is obtained for the Eulerian multi-fluid dynamics compared with Lagrangian description, Fig. 10.5, in the evaporating case. Therefore, even if crossings may appear locally in the flow, the multi-fluid method still predicts the global dynamics with a very good agreement with Lagrangian description.
These conclusions were already obtained in Chpt. 9, nevertheless the comparisons are provided in this chapter to validate the configuration before assessing fuel vapor mass fraction description. Furthermore, it allows to present a validation for inertial droplets in a gaseous field with a higher level of turbulence than in the axisymmetric case, and thus with higher occurrence of local droplet trajectory crossings. The monokinetic multi-fluid method is thus shown to be accurate in a more challenging context.

10.2 Multi-fluid gaseous fuel mass fraction prediction

The ability of the spray resolution method to describe fuel vapor mass fraction is a key element for combustion applications. The multi-fluid method, through its size phase space discretization, is able to predicts precisely the evaporation process. It is therefore expected to describe precisely the fuel vapor formation in the gaseous phase. The precision of the multi-fluid fuel mass fraction computations is assessed through comparisons with the prediction done by a Lagrangian DPS method. The framework adopted to conduct such comparisons is first recalled, as well as the computational methods used for the fuel

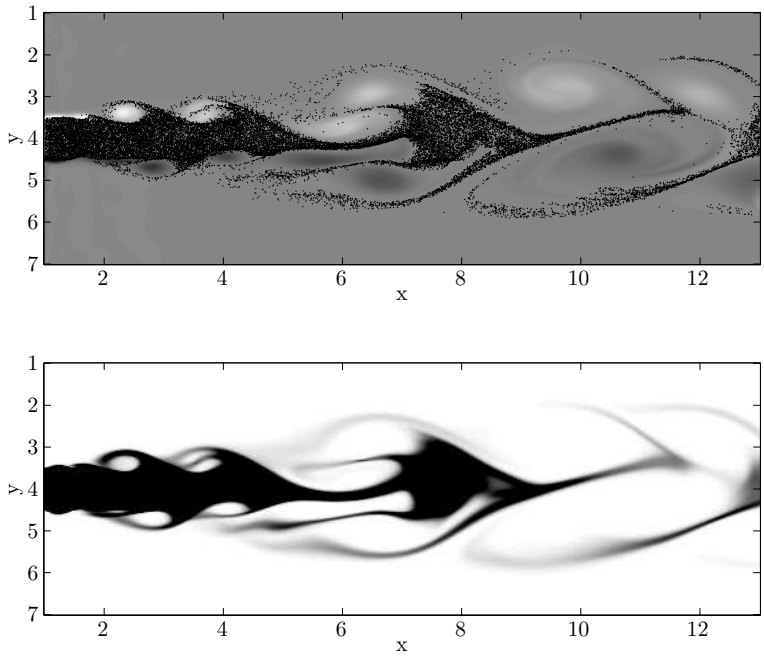


Figure 10.3 – ***Evaporating polydisperse spray****, whole size distribution, Stokes* 0.011 *to* 1.1 *corresponding to diameters* $D = 9\mu m$ *to* $d = 90\mu m$*, at time t=20: (top) Lagrangian particle positions with 30000 particles, (bottom) Eulerian multi-fluid monokinetic number density on a* $400 \times 200 \times 10$ *grid.*

vapor mass fraction resolution. Time resolved prediction of the fuel vapor by the multi-fluid methods are provided and evaluated.

10.2.1 Computing fuel vapor mass fraction in one-way coupling

The fuel mass fraction is computed through Eulerian and Lagrangian spray description. In order to isolate the intrinsic difference between these two descriptions of the fuel mass fraction, the gaseous phase should be the same for both computations. Therefore one does not want to take into account the influence of the fuel vapor on the gas phase. The fuel vapor is thus stored in a scalar transported and diffused by the gaseous flow, Eq. (1.108), but not influencing the gas thermodynamics, see Chpt. 1. A different scalar is used for each spray description. This framework, albeit quite artificial, is a first needed step to focus on the resolution methods, before studying their global impact on the whole flow computation. The numerical tools obtained coupling CORIA's ASPHODELE solver to MUSES3D, allows to describe, for the same gas phase computation, both descriptions of the fuel vapor mass fraction, see Chpt. 8. As far as resolution methods are concerned, the computation of the Lagrangian described fuel mass fraction is inserted within the global Runge Kutta method already used in ASPHODELE for gas and droplet resolution. On the other hand, the resolution of the Eulerian described fuel mass fraction is obtained by a splitting algorithm. The evolution due to the flow is computed with the global Runge Kutta scheme while the evolution due to Eulerian mass source term is given by a single Euler step. Details and justifications of these choices are given in Chpt. 8.

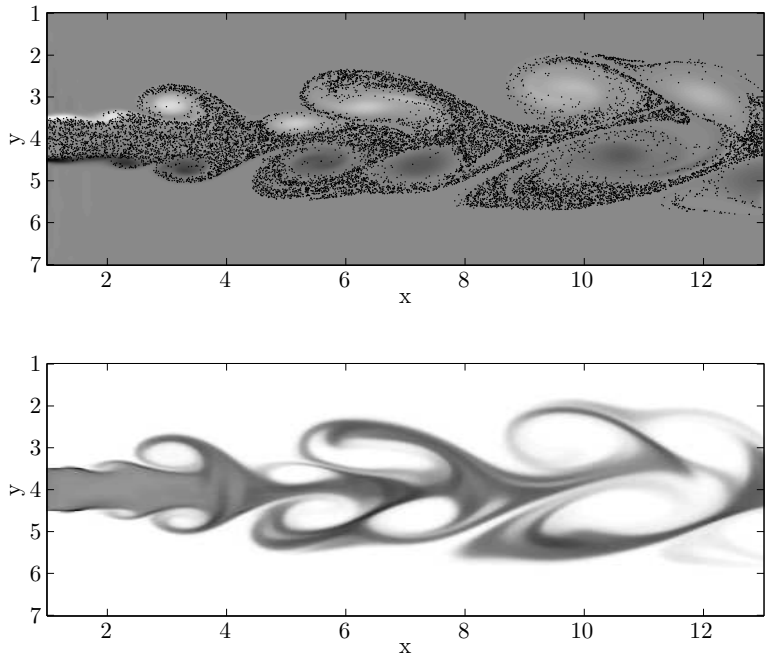


Figure 10.4 – ***Non-evaporating polydisperse spray, low inertia droplets****, Stokes* 0.011 *to* 0.12 *corresponding to diameters* $D = 9\mu m$ *to* $D = 30\mu m$*, at time t=20: (top) Lagrangian particle positions with 40000 particles, (bottom) Eulerian multi-fluid monokinetic number density on a* $400 \times 200 \times 5$ *grid.*

10.2.2 Eulerian Lagrangian vaporization comparison

The fuel vapor mass fraction obtained through multi-fluid spray description is shown in Fig. 10.6. Three times are represented, showing the evolution of the fuel vapor within the gas flow. The chosen method provide a physically consistent description, but no conclusion concerning precision can be drawn without comparing with the Lagrangian results. The gaseous fuel mass fractions obtained by both methods, along with the Eulerian number density and Lagrangian droplet positions are plotted in Fig. 10.7. Eulerian and Lagrangian fuel mass fractions alone are represented in Fig. 10.8 for a more precise comparison. This comparison underlines the efficiency of the Eulerian multi-fluid model in describing polydisperse evaporating sprays. Indeed, size distribution as well as size-velocity correlations are to be precisely described to obtain the correct fuel vapor distribution. Furthermore, as one can see in Fig. 10.8, the Eulerian description provides a smoother field than the Lagrangian one. It illustrates the difficulties arising when coupling the Lagrangian description of the liquid to the Eulerian description of gas and underlines the advantage of the spray Eulerian description for the liquid-gas coupling. These results are a first step towards combustion computations.

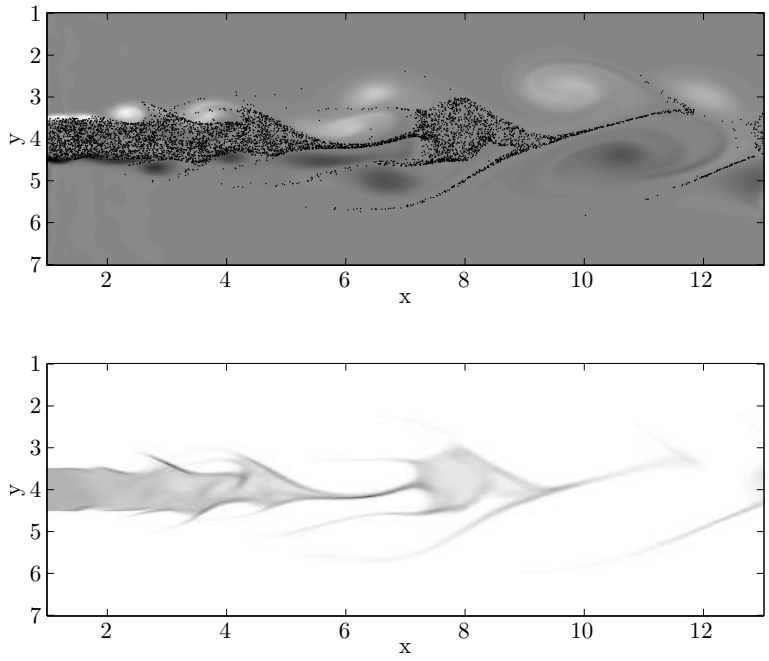


Figure 10.5 – ***Evaporating polydisperse spray, high-inertia droplets****, Stokes* 0.48 *to* 1.1 *corresponding to diameters* $D = 60$ μm *to* $d = 90$ μm*, at time t=20: (top) Lagrangian particle positions with 7000 particles, (bottom) Eulerian multi-fluid monokinetic number density with a* $400 \times 200 \times 10$ *grid.*

10.3 A first combustion case : coldflame computation

10.3.1 Coldflame framework

In order to assess a first combustion case, we study the gaseous combustion of the evaporated fuel. As it is aimed to conserve an approach devoted to spray resolution method evaluation, the combustion should not modify the gas phase thermodynamics. A peculiar framework is thus defined for combustion, given in details in Chpt. 1, and referred to as Coldflame. Therefore, as for the gaseous fuel mass fraction, the burnt gases and the oxidizer are defined in scalars that are not taken into account in the gas thermodynamics. Their evolution is given in Eq. (1.110).
The burnt gas mass fraction sets the temperature of the coldflame. The reaction is taken as a global one-step non-reversible reaction. The reaction rate is given by an Arrhenius law modified by GKAS technique, Eq. (1.115). The ignition is done at the chemical equilibrium.

10.3.2 Multi-fluid spray combustion

The fuel vapor fraction field obtained with the Eulerian multi-fluid description, Fig. 10.6, can be burnt using the coldflame framework. In this configuration, the only information available about the combustion process is the burnt gas mass fraction. This mass fraction is plotted in Fig. 10.9 for three different times. It is also represented in Fig. 10.10 (bottom),

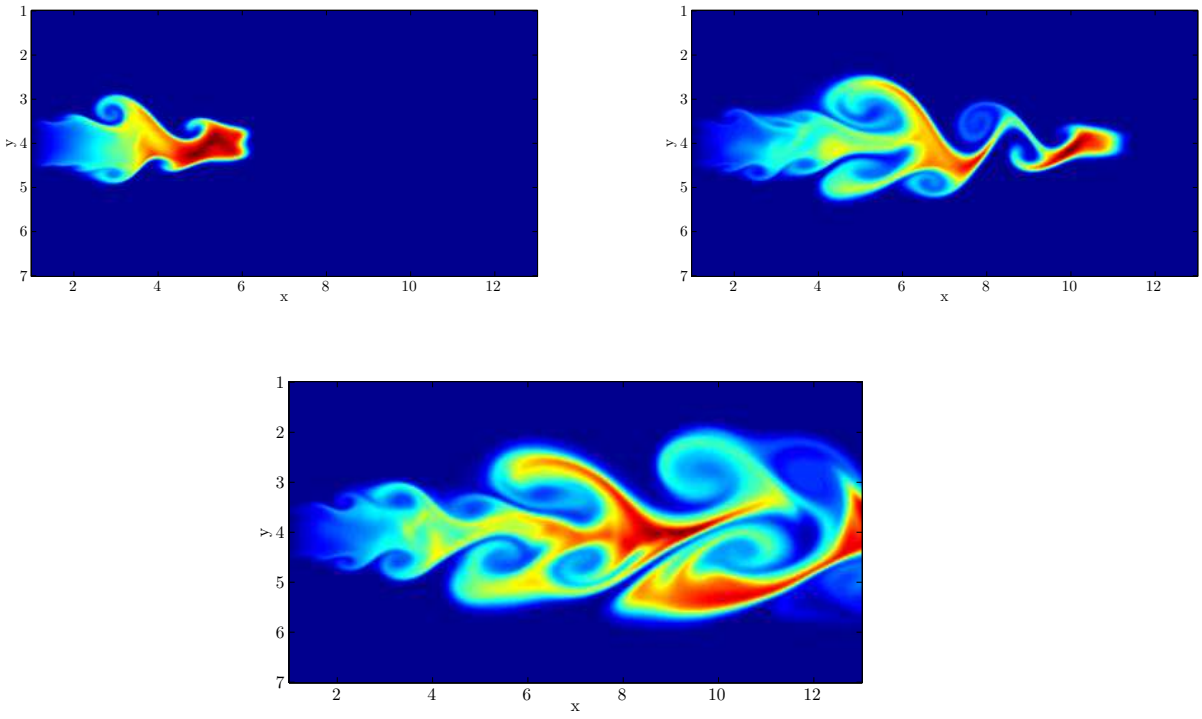


Figure 10.6 – *Gaseous fuel mass fraction obtained from multi-fluid computed spray evaporation, on a* $400 \times 200 \times 10$ *grid. Three times are plotted: (top-left)* $t = 5$*, (top-right)* $t = 10$*, (bottom)* $t = 20$*.*

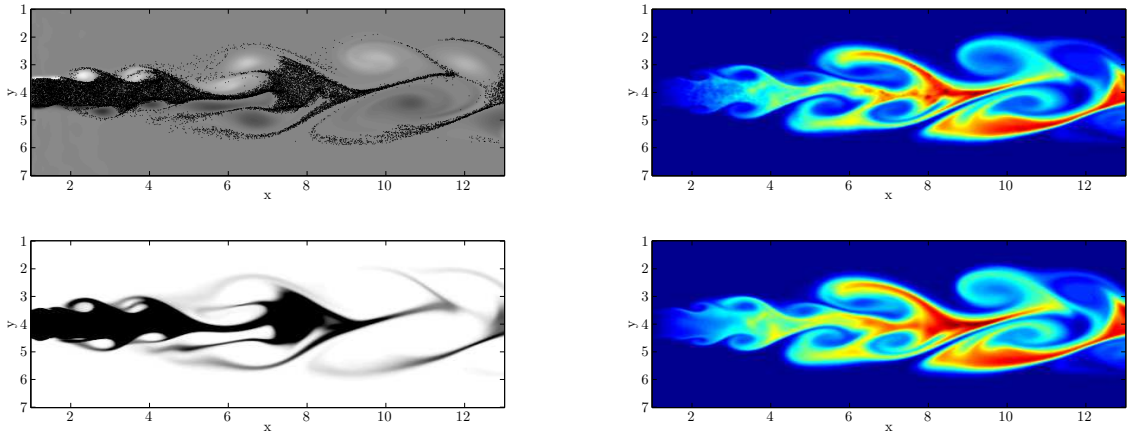


Figure 10.7 – *Lagrangian and Eulerian evaporating spray descriptions at time* $t = 20$*, obtained with a Lagrangian method with 30,000 droplets at the considered time and a Eulerian multi-fluid model with a* $400 \times 200 \times 10$ *grid. (Top-left) Lagrangian droplet positions, (bottom-left) Eulerian number density; (top-right) fuel vapor mass fraction from Lagrangian description, (bottom-right) fuel vapor mass fraction from Eulerian description.*

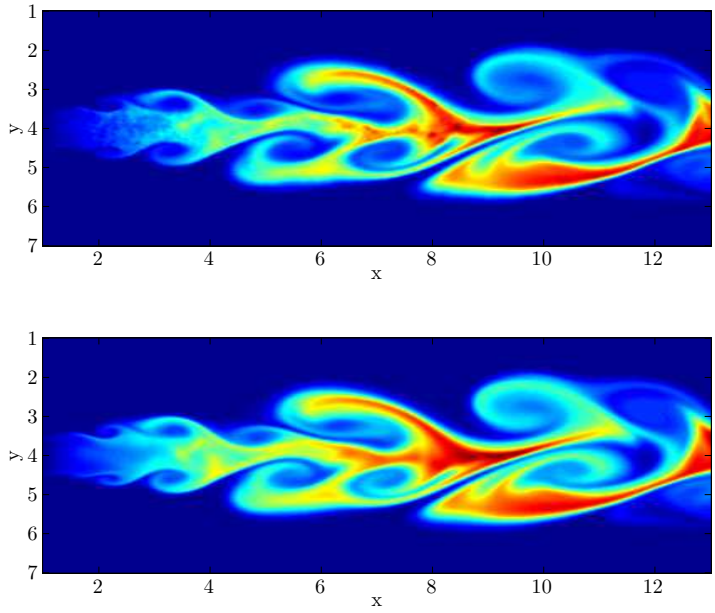


Figure 10.8 – *Comparison of the gaseous fuel mass fraction at time t=20, obtained from evaporation using (top) a Lagrangian method with 30,000 droplets at the considered time and (bottom) a Eulerian multi-fluid model with a* $400 \times 200 \times 10$ *grid.*

along with the resulting fuel vapor mass fraction, Fig. 10.10 (top), at time $t = 20$. This achievement provides a first step toward multi-fluid description of two-phase combustion applications, in multi-dimensional configurations.

The results obtained for the fuel vapor mass fraction description and for the Coldflame computation demonstrate three main points in the study of the multi-fluid model:

- the efficiency of the multi-fluid size distribution description as well as the size-velocity correlations;
- the global efficiency of the multi-fluid monokinetic formulation in turbulent jet configurations;
- the relevance of the multi-fluid model for combustion applications.

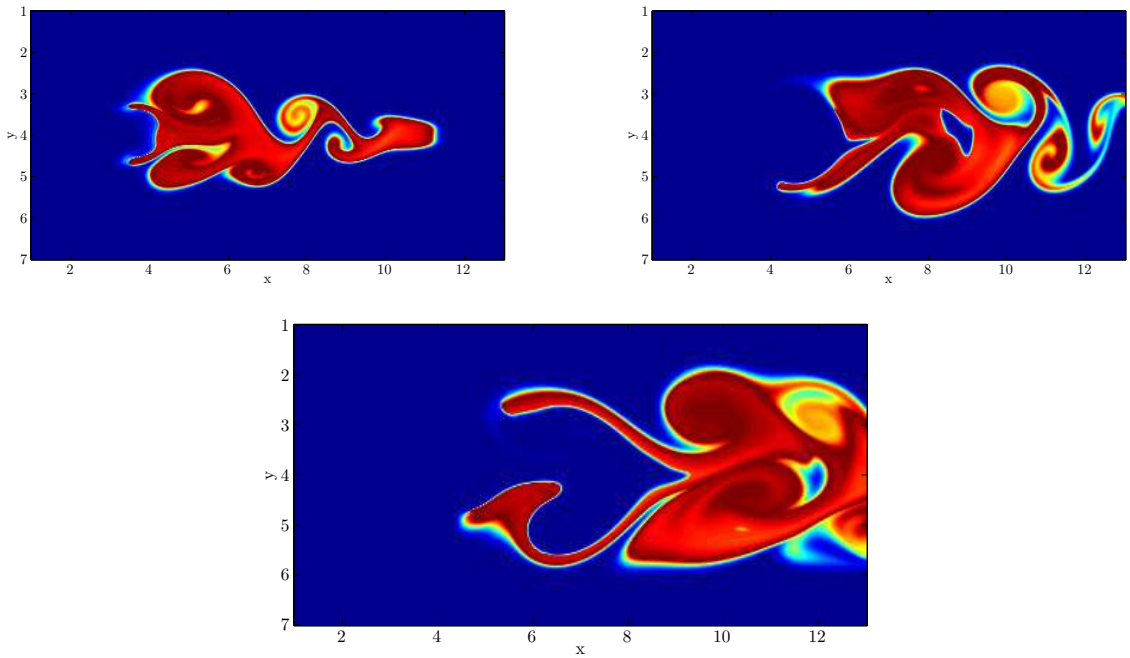


Figure 10.9 – *Burnt gas mass fraction obtained from multi-fluid coldflame computation on a* $400 \times 200 \times 10$ *grid. Three times are plotted: (top-left)* $t = 10$*, (top-right)* $t = 15$*, (bottom)* $t = 20$*.*

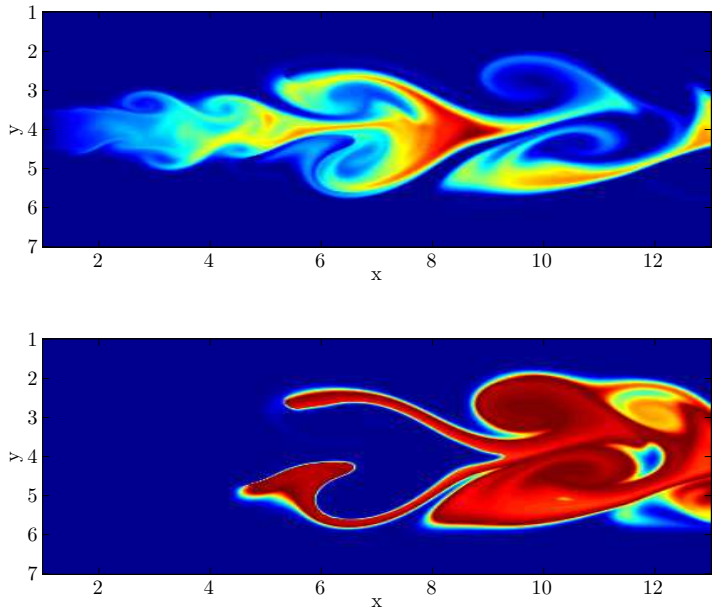


Figure 10.10 – *Coldflame computation, time* $t = 20$*,* 400×200 *grid: (top) gaseous fuel mass fraction obtained from multi-fluid evaporation (ten sections), (bottom) burnt gas mass fraction.*

Chapter 11

Frozen Homogeneous Isotropic Turbulence: a first 3-D case

Contents

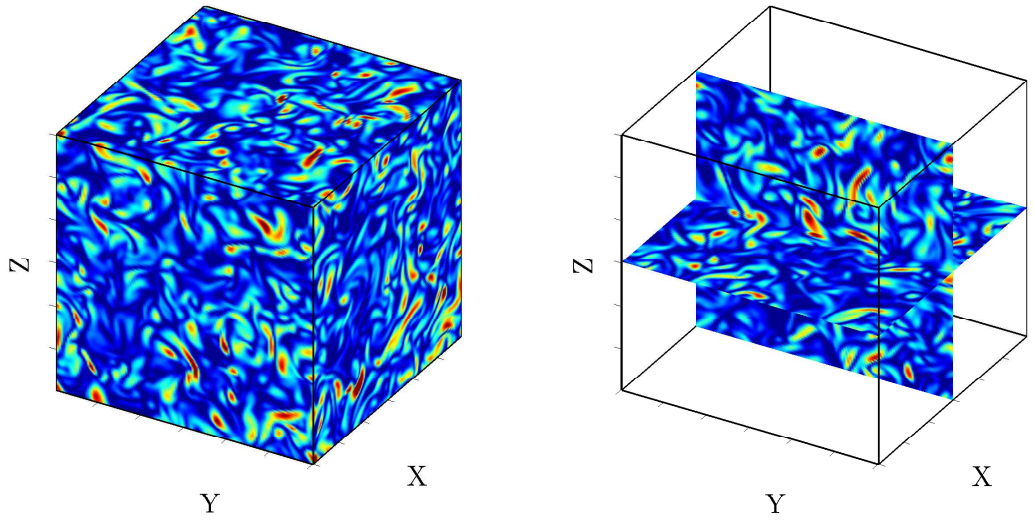
A 3-D computation with the multi-fluid method completes the evaluation of its applicability to industrial scale computations. The study of the dispersion of a polydispersed spray in a 3-D Homogeneous Isotropic Turbulence conducted in this chapter, represents a first step for 3-D multi-fluid evaluation.
The key point in this computation is to illustrate its feasibility and to provide a first analysis validating the multi-fluid description of the spray. This computation represents a challenge concerning scientific computing issues. Indeed, due to the size phase space discretization performed in the multi-fluid, this computation is 4-D. This computation is the outcome of the work done on the scientific computing issues in the multi-fluid framework, Part. III. The genericity obtained for the MUSES3D solver, Chpt. 6, allows to conduct easily its parallelization. Furthermore, the chosen domain decomposition strategy provides a great parallel efficiency, Chpt. 7, and makes the computations of the present chapter possible. The gas phase computation, as well as the Lagrangian one, used to provide a validation of the multi-fluid, were done at CORIA, (Reveillon and Demoulin 2007; Meftah 2008). Eulerian results are compared to the Lagrangian ones for the description of the spray dynamical equilibrium arising in HIT flows, (Reveillon and Demoulin 2007).
These recently obtained results provide an interesting basis to study the physics of the spray in HIT through a Eulerian size-conditioned description, that should be done in future works. The first 3-D configuration assessed with the multi-fluid model is provided in this chapter. It represents the first step toward the multi-fluid implementation in semi-industrial or industrial computational codes.

11.1 3-Dimensional DNS configuration

The Homogeneous Isotropic Turbulence configuration appears to be interesting to assess 3-D multi-fluid for three reasons:

- a spray dynamical equilibrium with the turbulence is reached, that corresponds to a stationary slip velocity standard deviation, (Reveillon and Demoulin 2007). This equilibrium leads to a stationary repartition of the spray density in the flow, conditioned by droplet size. This equilibrium is thus very interesting to compare the Eulerian and Lagrangian descriptions of the spray. This result can be linked with the work done in (Dufour and Villedieu 2005) showing the existence of an equilibrium velocity conditioned by size.
- HIT configuration provides very interesting points to assess Eulerian spray description, as crossing trajectories that can occur for inertial droplets. Furthermore, the study of the Eulerian velocity field of the spray conditioned by size is also a relevant issue. Although these points are not studied here, the required basis are provided.

Figure 11.1 – *Gas vorticity norm, obtained on a* 129^3 *grid with spectral resolution: (left) slice planes at the edges of the domain* ($x = x_{max}$, $y = y_{max}$, $z = z_{max}$), *(right) slice planes inside the domain* ($x = x_{max}/2$, $z = z_{max}/2$).



- Thanks to the use of a spectral solver, associated to a relatively low computational cost, the HIT configuration is also interesting for practical reasons.

11.1.1 Frozen Isotropic Homogeneous Turbulence

The properties of the statistically stationary turbulence in this configuration are described in Chpt. 4, and can be found in (Reveillon and Demoulin 2007). The statistically stationary state is reached with a deterministic forcing scheme keeping the mean kinetic energy at a prescribed level. The computational domain is cubic with a 3 mm edge. The computation is done on 129^3 points, with a single processor. Fig. 11.1 shows the gas vorticity through slice planes at the edge and inside the domain.

11.1.2 Dispersed liquid phase

A uniformly distributed spray is introduced in the turbulence several eddy turnover times after it reaches it stationary state. The spray is deposited with a zero initial velocity. The case studied in this chapter focuses on a non-evaporating spray. It will be essential to treat vaporizing case in order to take advantage of the multi-fluid size distribution description. One can note that there is no limitation to conduct a such case. The non-evaporating case was a necessary first case and the evaporating one will be addressed in a near future. As far as droplet models are concerned, a Stokes law is used for the drag force. No droplet interactions are taken into account and a one-way coupling is used. The particle inertia are described by their Stokes number based on the Kolmogorov length scale, (Wang and Maxey 1993), $\mathrm{St} = \tau_p/\tau_k$, τ_k being the characteristic time of the velocity fluctuations of the smallest structure, $\eta = 1.8 \times 10^{-5}$m.

11.2 Parallel multi-fluid computation for 3-D spray

The optimization of the MUSES3D platform through domain decomposition, Chpt. 7, allows to compute the dynamics of a spray in 3-D HIT for different droplet sizes with the multi-fluid model. The feasibility of this computation is shown and the used parallel setup discussed. The relevance of the multi-fluid method for parallel computations in a domain decomposition framework is illustrated here. This realization provides the starting point for further detailed physical analysis of Eulerian spray description.

11.2.1 Multi-fluid parallel setup

As far as the grid is concerned, the same Cartesian grid as the gas phase is chosen. One has to note that the gas is resolved through a cell-vertex finite-difference scheme, whereas the multi-fluid method uses a cell-center finite-volume method. Therefore the gas cell vertexes are taken as the multi-fluid cell centers. This choice was made for practical reason, in order to avoid spatial interpolation, nevertheless it is possible, see Chpt. 6, to use a more refined grid for the multi-fluid. A precise study of the refinement giving the best compromise between precision and efficiency for the multi-fluid should be conducted in the future.
A space domain decomposition is chosen for the multi-fluid, as advised by the study conducted in Chpt. 7. Therefore each process of the parallel computation has a domain $N_x^{\mathrm{p}} \times N_y^{\mathrm{p}} \times N_z^{\mathrm{p}} \times N_S$, where $N_x^{\mathrm{p}} \times N_y^{\mathrm{p}} \times N_z^{\mathrm{p}}$ is the number of grid points of the process sub-domain and N_S is the number of multi-fluid size sections.
The parallel computations are distributed memory computations using MPI communications, Chpt. 7. The best efficiency for MUSES3D solver in this configuration is obtained, as established in Chpt. 7, for 100 MPI parallel processes. This setup has been obtained with MADNESS EM2C cluster, described in Chpt. 7. The domain decomposition minimizing the communication time is obtained for $(N_{\mathrm{p}x}, N_{\mathrm{p}y}, N_{\mathrm{p}z}) = (5, 5, 4)$, where $N_{\mathrm{p}i}$ is the number of processes in the i^{th} direction. This is due to the InfiniBand connections of the cluster favoring a high number of few data containing communications.

11.2.2 Eulerian size-conditioned spray dynamics

The Eulerian multi-fluid description of the spray dynamics are presented in this section for two Stokes number, based on the Kolmogorov length scale:

- St = 1.05, corresponding to droplet with diameter $D = 48\mu$m;
- St = 0.17, corresponding to droplet with diameter $D = 20\mu$m.

These two different inertia allow to study a spray ejected from the center core and segregated in weak vorticity areas, (Reveillon and Demoulin 2007). They are thus well suited for robustness evaluation of the multi-fluid method. Indeed high density regions, as well as vacuum, are created, that represent a challenging issue for a Eulerian method. Higher Stokes number are not tackled here since it was shown in (Reveillon and Demoulin 2007) that, for Stokes number greater than unity, the droplets are inertial enough to cross high-vorticity areas, leading to a less segregated spray. In this case droplet trajectory crossings

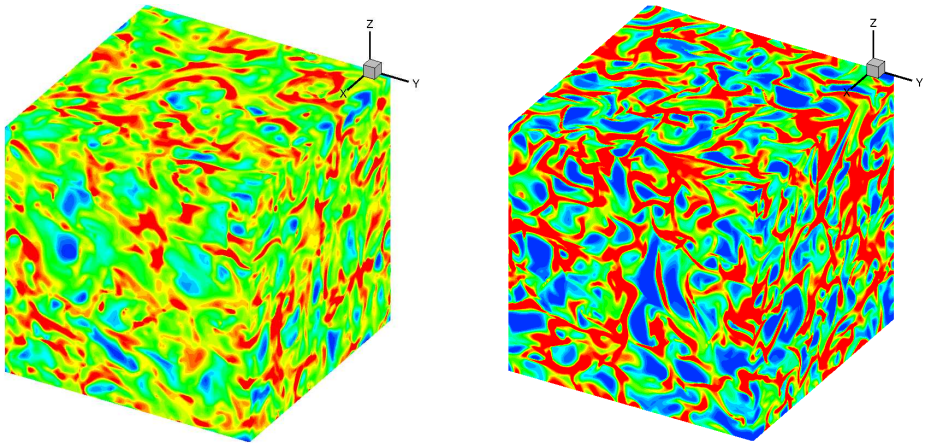


Figure 11.2 – *Eulerian spray number density obtained with the multi-fluid method on a* 128^3 *Cartesian grid. Slice planes at the edges of the domain* ($x = x_{max}$, $y = y_{max}$, $z = z_{max}$) *for two Stokes number: (left)* $St = 0.17$, *(right)* $St = 1.05$.

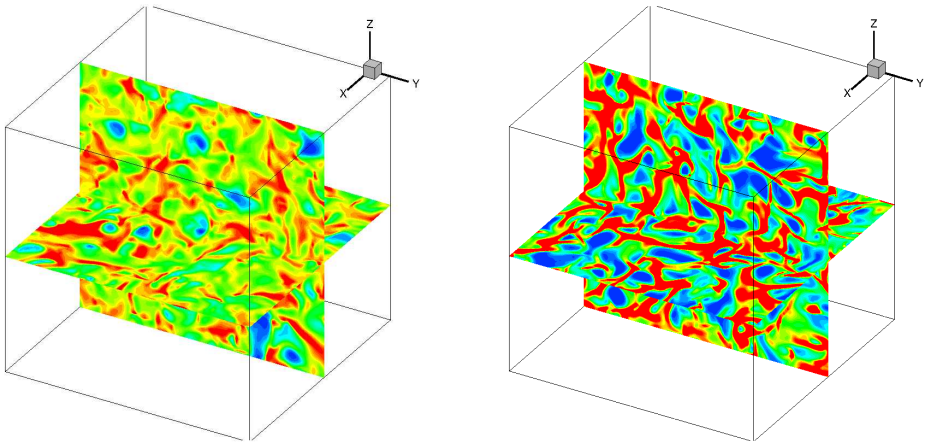


Figure 11.3 – *Eulerian spray number density obtained with the multi-fluid method on a* 128^3 *Cartesian grid. Slice planes inside the domain* ($x = x_{max}/2$, $z = z_{max}/2$) *for two Stokes number: (left)* $St = 0.17$, *(right)* $St = 1.05$.

have a strong impact on the spray repartition and the monokinetic assumption of the multi-fluid might not allow to describe it. A precise study extending the notion of critical Stokes number, introduced in Chpt. 3, to this 3-D HIT configuration, would be needed to analyze precisely the impact of the monokinetic assumption on the spray modeling. Nevertheless, our aim here is only to provide a first realization of 3-D multi-fluid spray computation as well as a first validation. The number density of the spray is plotted in planes at the edges and inside the domain, in Fig. 11.2 and Fig. 11.3, respectively, to provide a global comprehension of the spray repartition. Both Stokes number St = 0.17 and St = 1.05 are plotted in each figure. Figures 11.2 and 11.3 show, as expected, two different dynamics, with a higher segregation for the Stokes St = 1.05, and thus with high concentration regions and vacuum.

Interesting perspectives can be thought of from these results. Indeed, they provide the basis for physical studies on Eulerian description of spray ejection in 3-D vortices. The analysis of the Eulerian velocity field, will enable to assess the existence of an equilibrium velocity variety, introduced in (Dufour 2005). The study of this attracting velocity field will provide

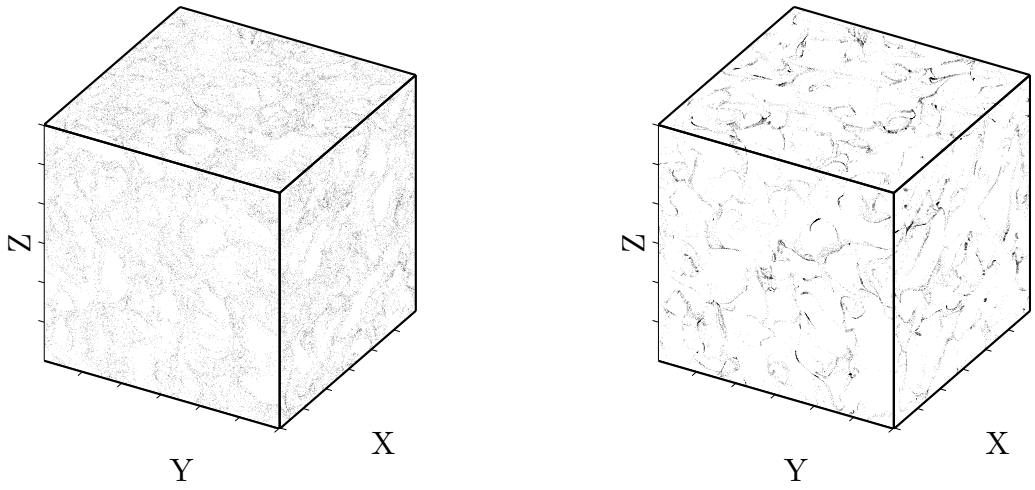


Figure 11.4 – *Lagrangian droplet positions obtained by a DPS with* 2150000 *droplets. Slice planes at the edges of the domain ($x = x_{max}$, $y = y_{max}$, $z = z_{max}$) for two Stokes number: (left) $St = 0.17$, (right) $St = 1.05$.*

comprehension of droplet ejection dynamics. Furthermore, droplet trajectory crossings can be studied, in order to provide a Stokes criteria to foresee their formation, as done in 2-D case in Chpt. 3 and Chpt. 4. This study could also assess the interest of Eulerian method describing out of equilibrium velocity distribution, (Fox 2008; de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009; Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009).

11.3 3-D multi-fluid Lagrangian validation

This first 3-D spray computation with multi-fluid has to be completed by a first validation. Global qualitative comparisons with a Lagrangian description of the spray are provided in order to assess the multi-fluid description in the range of Stokes number studied.

11.3.1 Lagrangian reference

A Lagrangian DPS computation with two Million droplets uniformly distributed in the domain is used to perform comparisons. This computation has been done at CORIA laboratory, (Meftah 2008; Reveillon and Demoulin 2007). The spray dynamical equilibrium is shown through the droplet positions, for the two considered Stokes, in Fig. 11.4 and Fig. 11.5. The representation is done in the same planes as for Eulerian density plots, Fig. 11.2 and Fig. 11.3.

11.3.2 Eulerian Lagrangian comparisons

First qualitative comparisons are provided in this paragraph. To assess the multi-fluid description of the size-conditioned dynamics, Eulerian density fields are compared to La-

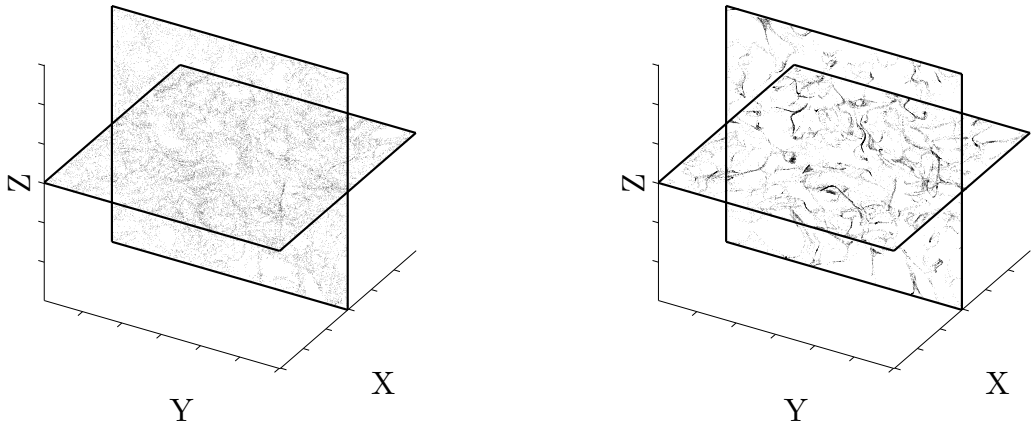


Figure 11.5 – *Lagrangian droplet positions obtained by a DPS with* 2150000 *droplets. Slice planes inside the domain ($x = x_{max}/2$, $z = z_{max}/2$) for two Stokes number: (left) $St = 0.17$, (right) $St = 1.05$.*

grangian droplet positions. This type of qualitative comparisons has already been introduced and justified in Chpt. 9. This procedure is applied in the planes $x = x_{max}/2$, Fig. 11.6 to Fig. 11.8, and $z = z_{max}/2$, Fig. 11.9 to Fig. 11.11. In the chosen inertial range, the spray is ejected from the vortex cores and accumulated in low vorticity areas. In order to link the spray dispersion given by both methods, to the gas vorticity structure, the square norm of the gas vorticity is given in Fig. 11.6 and Fig. 11.9, for the planes $x = x_{max}/2$ and $z = z_{max}/2$, respectively.

The comparisons are provided for low Stokes droplets, St = 0.17, in Fig. 11.7 and Fig. 11.11, and for higher Stokes droplets, St = 1.17, in Fig. 11.8 and Fig. 11.11.

Qualitative comparisons between both approach can be done focusing on the vacuum zones description. These zones correspond to the gas vortex cores, that can be identified from the vorticity representation provided in Fig. 11.6 and Fig. 11.9. The repartition of these vacuum zones obtained by the classical Lagrangian method is very precisely reproduced by the multi-fluid on the different planes chosen, see Fig. 11.7 and Fig. 11.10. Furthermore, the evolution of droplet repartition with inertia is very well captured by the multi-fluid. Indeed, the Eulerian density fields for higher Stokes number still present very good agreeement with the Lagrangian droplet repartitions, see Fig. 11.8 and Fig. 11.11.

It is obvious that these comparisons are not enough to characterize precisely the behavior of the Eulerian multi-fluid model in this configuration. Nevertheless, the aim of this chapter is above all to illustrate the efficiency of the developments realized in this work, Chpt. 7, to obtain an implementation of the multi-fluid method in a parallel framework. This point is made by the ability to compute this 3-D HIT configuration, that represents a first step toward more complex 3-D configurations, as 3-D jet that are of great interest for multi-fluid applications. The comparisons conducted with a Lagrangian reference are very encouraging as far as the precision of the multi-fluid method in such configurations is concerned.

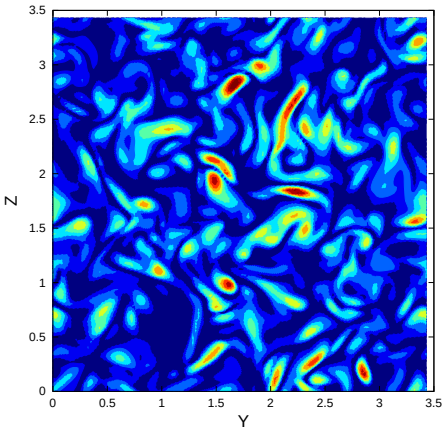


Figure 11.6 – *Gas vorticity square norm in the $y-z$ plane at $x = x_{max}/2$. The gaseous turbulence has reached its stationary state.*

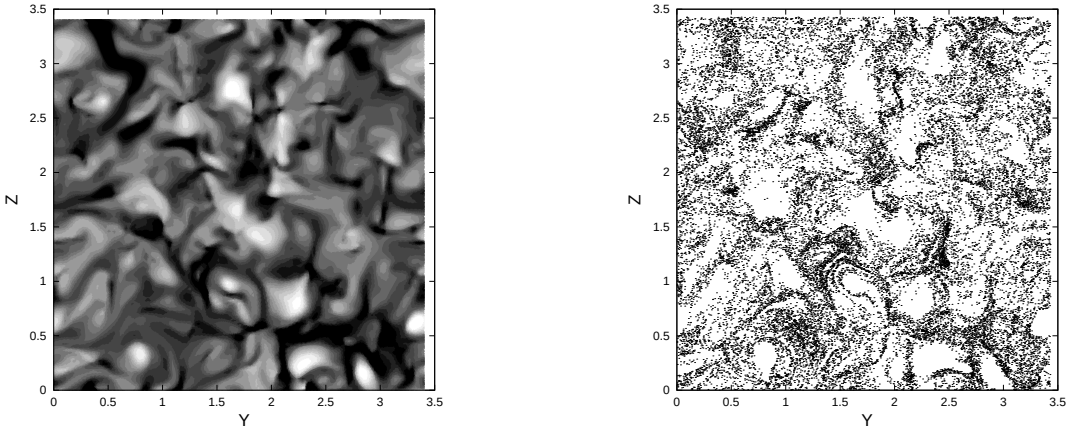


Figure 11.7 – *Eulerian Lagrangian comparisons in the $y-z$ plane at $x = x_{max}/2$ for low inertia droplets, $St = 0.17$: (left) Eulerian multi-fluid number density, (right) Lagrangian droplet positions.*

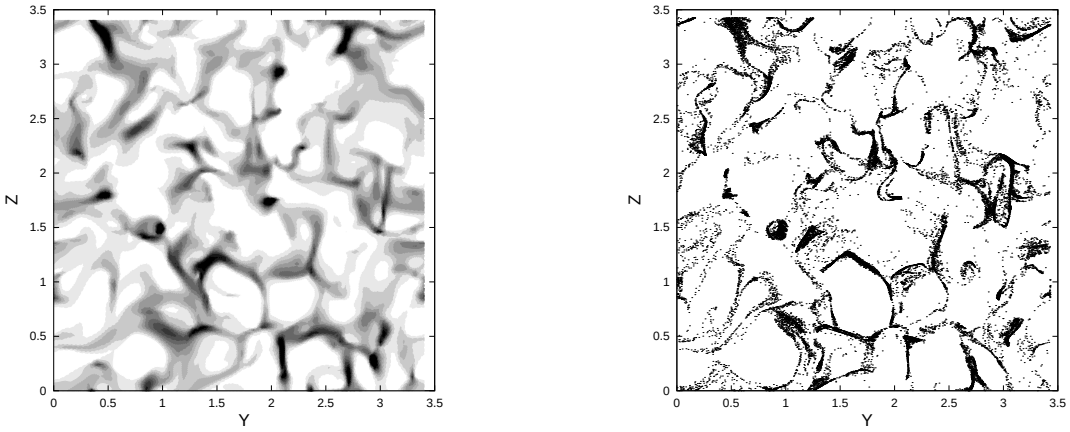


Figure 11.8 – *Eulerian Lagrangian comparisons in the $y-z$ plane at $x = x_{max}/2$ for higher inertia droplets, $St = 1.05$: (left) Eulerian multi-fluid number density, (right) Lagrangian droplet positions.*

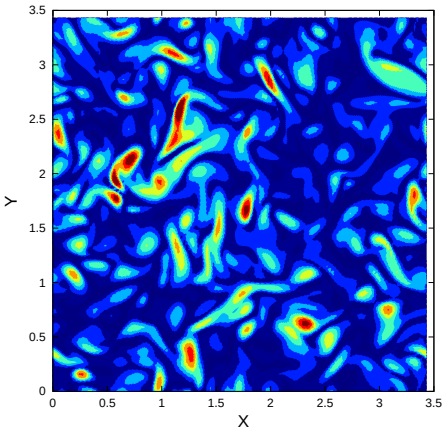


Figure 11.9 – *Gas vorticity square norm in the $x-y$ plane at $z = z_{max}/2$. The gaseous turbulence has reached its stationary state.*

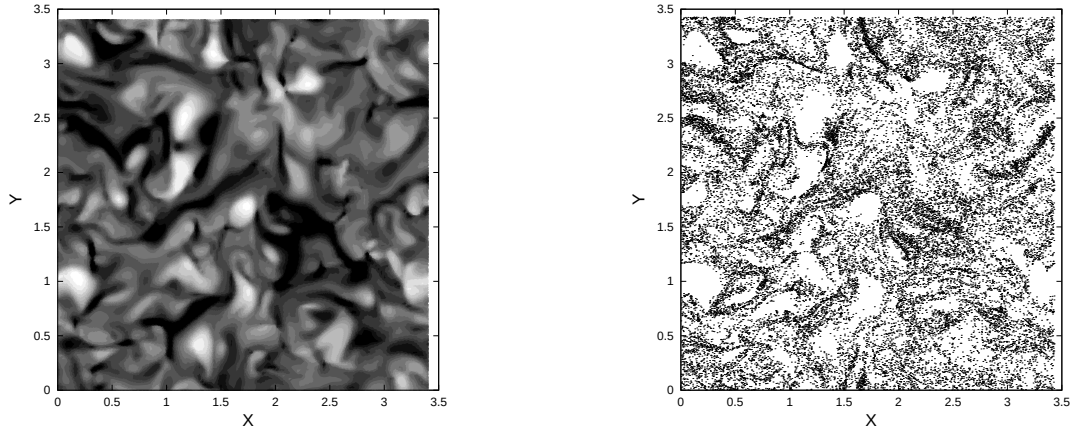


Figure 11.10 – *Eulerian Lagrangian comparisons in the $x-y$ plane at $z = z_{max}/2$ for low inertia droplets, $St = 0.17$: (left) Eulerian multi-fluid number density, (right) Lagrangian droplet positions.*

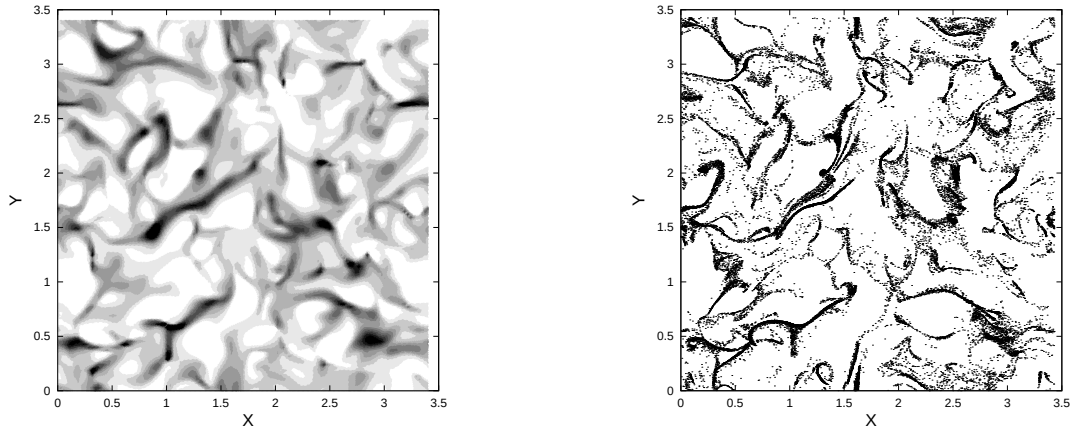


Figure 11.11 – *Eulerian Lagrangian comparisons in the $x-y$ plane at $z = z_{max}/2$ for higher inertia droplets, $St = 1.05$: (left) Eulerian multi-fluid number density, (right) Lagrangian droplet positions.*

Conclusion

The study conducted in this work combines mathematical, numerical and physical analysis of the multi-fluid method. The analysis of this Eulerian polydispersed spray description through these different angles allows to provide in this work:

- specific numerical schemes design and analysis. The mathematical study conducted allows to design robust numerical schemes. These schemes deal with the main difficulties arising in spray simulation such as vacuum zones and high density regions. Furthermore, the developed numerical methods can resolve the singularities arising due to the monokinetic assumption, the so-called $\delta-$shock. The provided numerical schemes ensure second order accuracy in space and time.
- Numerical tools devoted to spray methods evaluation are developed. They provide an original implementation of the multi-fluid method, combining a high genericity level and an important computational efficiency in a parallel framework. These tools allow to compute 2-D, 2-D axisymmetric and 3-D configurations.
- Provided models and methods are validated. The physics of the multi-fluid model is assessed through comparisons with numerical and experimental results. These comparisons are the starting point for a physical analysis of spray dispersion through a Eulerian size-conditioned description.

This multidisciplinary approach leads to the first Eulerian, size-conditioned, spray dynamics computations in configurations close to industrial concerns in terms of computational complexity, using an efficient parallel solver. It shows the multi-fluid method relevance for industrial scale simulations through two main points:

- the feasibility of multi-fluid computations in reasonably realistic configurations. This point lies on the scheme robustness and on the computational efficiency of their implementation;
- the precision obtained in this context, established through numerical and experimental validations.

This major achievement comes with four important novelties:

- time-resolved comparisons between Euler/Lagrange and Euler/Euler descriptions are provided in a DNS context, a key element to derive closure for LES simulations. These comparisons are done at two levels. Precise quantitative comparisons between the multi-fluid and a DSMC Lagrangian method are conducted in 2-D vortical flows. In the framework of more realistic flows, as 2-D jets or 3-D configurations, more global qualitative comparisons are chosen. A DPS Lagrangian approach, leading to a refinement closer to the one found in industrial configurations, is used. An important

novelty of this analysis comes from its time resolution. Indeed, dynamical Eulerian/Lagrangian comparisons are provided, where, for the first time, instantaneous fields are compared.

- The same type of time-resolved, size-conditioned comparisons are provided with experimental data. The size-conditioned description of the dynamics is a novel element, as much from the experimental as from the numerical point of view, where a polydispersed Eulerian description is used.
- An operational evaluation of Eulerian and Lagrangian approaches, focused on precision and computational cost is conducted. For a given resolution level, the Eulerian multi-fluid method presents costs equivalent to the Lagrangian ones. Compared to a coarser Lagrangian simulation, closer to industrial concerns, the multi-fluid method is more expensive, but provides more information. Furthermore, this extra cost will not appear in a parallel framework, given the high efficiency of the multi-fluid in this context and the difficulties encountered in Lagrangian parallel simulations.
- Finally, an academic solver combining a high parallel efficiency and a high level of genericity has been developed. It provides a parallel efficiency close to unity using Gustafson's speed up definition. The high genericity provided has been illustrated in (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009; Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009), where different types of multi-fluid methods have been assessed in relatively complex jet configurations. This numerical tool provides a spray resolution devoted framework, allowing detailed dynamics evaluation of evaporating sprays, possibly with combustion. This solver demonstrates the feasibility of multi-fluid model in more complex configurations, and illustrates its relevance for industrial scale computations.

One can note that mathematical and physical modeling issues, as well as scientific computing issues, conditioning the computational cost, are necessary in order to move toward an industrial tool providing **Eulerian**, **numerically robust**, **polydispersed** spray description.

The multi-fluid numerical and experimental characterization were done in the framework of a Young Investigator Award from the French Ministry of Research (New Interfaces of Mathematics - M. Massot, 2003-2006), and of an ANR (National Research Agency - France) Young Investigator Award (M. Massot, 2006-2009, N: ANR-05-JC05-42263). The operational evaluation of Eulerian and Lagrangian approaches was done thanks to the support of European Commission through the project "Toward Innovative Methods for Combustion Prediction in Aero-Engines" (TIMECOP-AE, project N: AST5-CT-2006-030828).

As far as the perspectives of this work are concerned, three types of extension can be thought of. The first one represents a direct continuation of the work provided in this study, developing the physical analysis of evaporating spray dispersion and combustion. The second one is related to the outbreak of new types of multi-fluid methods. Finally the third one concerns the evolution one can expect for the multi-fluid method, i.e., its use in LES of industrial devices.

- The direct extension of the provided work concerns the physical analysis of evaporating and burning spray dynamics, using the Eulerian multi-fluid method. Indeed, this work is devoted to the evaluation of the multi-fluid method, in a simplified frame-

work. A more global framework should be used to analyze the physics of the spray. Given the genericity kept throughout this work, more complex evaporation and heat transfer models are at present available in the numerical tools provided. Therefore the physics of evaporation and combustion of a fuel spray could be studied through a Eulerian method. Two-way coupling should be used in this context. Furthermore, the 3-D computations achieved in the case of the Homogeneous Isotropic Turbulence can lead to physical studies of the interaction between the spray and the 3-D turbulent eddies. The Eulerian description, giving access for instance to a spray velocity field, can bring in a refined physical comprehension. The behavior of the multi-fluid method in a 3-D jet configuration should also be assessed.

- The second point is related to the evolution of the multi-fluid method. Indeed new high order methods of moments are studied at present by Damien Kah, thanks to a PhD grant of IFP, at EM2C Laboratory. On the one side, high order method of moments for the size phase space, leading to an improved evaporation description are developed, (Massot, Laurent, Kah, and de Chaisemartin 2009). On the other side, high order methods for the velocity phase space, allowing to describe droplet trajectory crossings, recently outbroke (Fox 2008), and have been extended to the multi-fluid framework, (de Chaisemartin, Freret, Kah, Laurent, Fox, Reveillon, and Massot 2009; Freret, de Chaisemartin, Laurent, Vedula, Fox, Thomine, Reveillon, and Massot 2009).
- Finally, the last perspective concerns the use of the multi-fluid method in the framework of industrial device LES. Two main issues have to be addressed in this context and are currently work in progress.
  - The first point concerns the use of the multi-fluid method for LES. In this context, size-conditioned closure have to be provided, using the first results presented in (Reveillon, Péra, Massot, and Knikker 2004). Furthermore, mathematical studies of the resulting equations could provide the elements to extend the developed numerical schemes to this framework. The main difference comes from the pressure arising in the momentum equation, vacant in the DNS case due to the monokinetic assumption.
  - The second point is related to the complex geometry arising in this framework. Unstructured extension of the provided mesh is thus to consider. In order to implement a specific numerical scheme in a semi-industrial solver such as AVBP, co-developed at CERFACS and IFP, the evolution of the cell-centered kinetic scheme provided in this work, to a cell-vertex formulation should be considered.

References

Abramzon, B. and W. Sirignano (1989). Droplet vaporization model for spray combustion calculations. *Int. J. Heat Mass Transfer 32*, 1605–1618. 16, 19, 20, 21

Achim, P. (1999). *Simulation de collisions, coalescence et rupture de gouttes par une approche lagrangienne: application aux moteurs à propergol solide*. Ph. D. thesis, Faculté des Sciences de l'Université de Rouen. 23

Allaire, G., S. Clerc, and S. Kokh (2002). A five equation model for the simulation of interfaces between compressible fluids. *J Comp Phys 181*(2), 577–616. xix

Amsden, A., P. J. O'Rourke, and T. D. Butler (1989). Kiva II, a computer program for chemically reactive flows with sprays. Technical Report LA-11560-MS, Report Los Alamos National Laboratory, Los Alamos, New Mexico. xx, 34, 35

Babinsky, E. and P. E. Sojka (2002). Modeling drop size distribution. *Progress in Energy and Combustion Science 28*, 303–329. 11, 47

Bardos, C., F. Golse, and D. Levermore (1991). Fluid dynamic limits of kinetic equations. i. formal derivations. *Journal of Statistical Physics 63*. xxiv

Bell, J. B. and D. M. Marcus (1992). A second-order projection method for variable-density flows. *J. Comp. Phys. 101*, 71–97. 217

Birbaud, A. L., D. Durox, S. Ducruix, and S. Candel (2007). Dynamics of free jets submitted to upstream acoustic modulations. *Physics of Fluids 19*(1). 133, 134

Bird, G. A. (1994). Molecular gas dynamics and the direct simulation of gas flows. *Oxford Science Publications 42*. xxii, 36

Bobylev, A. V. and T. Ohwada (2001). The error of the splitting scheme for solving evolutionary equations. *Applied Mathematics Letters 14*, 45–48. 78

Boileau, M. (2007). *Simulation aux grandes échelles de l'allumage diphasique des foyers aéronautiques*. Ph. D. thesis, Institut National Polytechnique de Toulouse. xviii, 34, 54, 184

Borghi, R. and M. Destriau (1995). *La combustion et les flammes*. Technip. xix

Bouchut, F. (1994). On zero pressure gas dynamics. In *Advances in kinetic theory and computing*, pp. 171–190. River Edge, NJ: World Sci. Publishing. 68, 69, 71

Bouchut, F., S. Jin, and X. Li (2003). Numerical approximations of pressureless and isothermal gas dynamics. *SIAM J. Num. Anal. 41*, 135–158. 71, 80, 82, 83, 84, 87, 88, 89, 91, 93

Candel, S. (1990). *Mécanique des fluides* (second ed.). Paris: Dunod. 6, 8

Candel, S. (2002). Combustion dynamics and control: Progress and challenges. *Proceedings of the Combustion Institute 29*(1), 1–28. xvii

Candel, S., F. Lacas, N. Darabiha, and J.-C. Rolon (1999). Group combustion in spray flames. *Multiphase Science and Technology 11*, 1–18. xix

Canneviere, K. (2003). *Simulation numérique directe de la combustion turbulente diphasique: Application à l'étude de la propagation et de la structure des flammes*. Ph. D. thesis, INSA de Rouen. 20, 30

Cary, J. R., S. G. Shasharina, J. C. Cummings, J. V. W. Reynders, and P. J. Hinker (1997). Comparison of C++ and Fortran 90 for object-oriented scientific programming. *Computer Physics Communications 105*, 20–36. 150, 151

Cercignani, C. (1988). *The Boltzmann equation and its applications*. New York: Springer-Verlag. 21, 22

Cercignani, C., R. Illner, and M. Pulvirenti (1994). *The mathematical theory of dilute gases*. New York: Springer-Verlag. 21

Chanteperdrix, G., P. Villedieu, and J. P. Vila (2002). A compressible model for separated two-phase flows computations. In *ASME Fluids Engineering Division Summer Meeting*, Number 31141, Montreal. 44

Chergui, J., I. Dupays, D. Girou, and P.-F. Lavallée (2006). Extensions à message Passing Interface (MPI-2). Technical report, French Institution IDRIS Course. Available at http://www.idris.fr/data/cours/parallel/mpi/choix_doc.html. 200, 203

Chergui, J., I. Dupays, D. Girou, S. Requena, and P. Wautelet (2006). Message Passing Interface (MPI-1). Technical report, French Institution IDRIS Course. Available at http://www.idris.fr/data/cours/parallel/mpi/choix_doc.html. 200

Chorin, A. J. (1968). Numerical solution of the Navier-Stokes equations. *Math. Comp. 22*, 745–762. 217

Clift, R., J. R. Grace, and M. E. Weber (1978). *Bubbles, Drop and Particles*. Academic Press, New York. 15, 21

Corde, P. and H. Delouis (2008). Les apports de fortran 2003. Technical report, French Institution IDRIS Course. Available at http://www.idris.fr/data/cours/lang/fortran/choix_doc.html. 148, 151

Corde, P., H. Delouis, and J.-M. Dupays (2008). Cours fortran 95. Technical report, French Institution IDRIS Course. Available at http://www.idris.fr/data/cours/lang/fortran/choix_doc.html. 148, 151

Crow, S. and F. Champagne (1971). Orderly structure in jet turbulence. *J. Fluid Mech 48*, 547. 132, 133

Crowe, C. T., M. P. Sharma, and D. E. Stock (1977). The particle-source-in cell (psi-cell) model for gas-droplet flows. *J. Fluids Eng.*, 325–332. 219

Crowe, C. T., M. Sommerfeld, and Y. Tsuji (1998). *Multiphase Flows with droplets and particles*. Boca Raton, Florida: CRC Press. 13, 16, 35

de Chaisemartin, S., L. Freret, D. Kah, F. Laurent, R. O. Fox, J. Reveillon, and M. Massot (2009). Eulerian models for turbulent spray combustion with polydispersity and droplet crossing : modeling and validation. Proceedings of the Summer Program 2008, Center for Turbulence Research, Stanford University. xxiv, xxvi, 50, 51, 55, 96, 118, 146, 238, 252, 258, 259

de Chaisemartin, S., F. Laurent, L. Fréret, M. Massot, A.-L. Birbaud, C. Lacour, S. Ducruix, and D. Durox (2007). Pulsated free jets with spray injection : Eulerian Multi-Fluid modelling and simulation versus experimental measurements. In *Proceedings of the International Conference on Multiphase Flows, ICMF07, Leipzig*. 132, 133, 134, 135, 137, 138, 139

de Chaisemartin, S., F. Laurent, M. Massot, and J. Reveillon (2006). LES multi-fluid formulation and applied mathematical issues. Technical report, CRSA-EM2C, Deliverable 1.3.1a, European Project TIMECOP-AE. 104

de Chaisemartin, S., F. Laurent, M. Massot, and J. Reveillon (2007). Evaluation of Eulerian Multi-Fluid versus Lagrangian methods for the ejection of polydisperse evaporating sprays by vortices. In *Proceedings of the International Conference on Multiphase Flows, ICMF07, Leipzig*. 138

de Chaisemartin, S., F. Laurent, M. Massot, and J. Reveillon (2008a). Complete report with comparisons of Lagrangian and Eulerian formulation from both the physical and the numerical point of view. Technical report, CRSA-EM2C, Deliverable 1.3.1b, European Project TIMECOP-AE. 228

de Chaisemartin, S., F. Laurent, M. Massot, and J. Reveillon (2008b). Evaluation of Eulerian Multi-Fluid versus Lagrangian methods for the ejection of polydisperse evaporating sprays by vortices. *Journal of Computational Physics*. Submitted, available on HAL http://hal.archives-ouvertes.fr/hal-00169721/. 104

Decyk, V. K., C. D. Norton, and B. K. Szymanski (1997). Expressing object-oriented concepts in Fortran90. In *ACM Fortran Forum*, Volume 16. 148

Delhaye, J. M. and J. M. Achard (1977). On the use of averaging operators in two-phase flow modeling. *ASME, 1*, 289–332. xix

Demoulin, F., G. Blokkeel, A. Mura, P. Beau, and R. Borghi (2007). Modeling of liquid atomization as a turbulent flow with high density fluctuations. *Atomization and Sprays 17*, 298–313. xx

Descombes, S., T. Dumont, V. Louvet, M. Massot, F. Laurent, and J. Beaulaurier (2009). Operator splitting techniques for multi-scale reacting waves and application to low mach number flames with complex chemistry: theoretical and numerical aspects. *SIAM J. Scient. Comp.*. submitted for publication. 224

Descombes, S. and M. Massot (2004). Operator splitting for nonlinear reaction-diffusion systems with an entropic structure: singular perturbation and order reduction. *Numer. Math. 97*(4), 667–698. 78, 79

Desjardins, O., R. O. Fox, and P. Villedieu (2008). A quadrature-based moment method for dilute fluid-particle flows. *Journal of Computational Physics 227*(12), 6313–6350. 117

Drew, D. A. and S. L. Passman (1999). Theory of multicomponent fluids. *Applied Mathematical Sciences 135*. 44

Druzhinin, O. A. and S. E. Elghobashi (1999). A Lagrangian-Eulerian mapping solver for direct numerical simulation of bubble-laden turbulent shear flows using the two-fluid formulation. *J. Comput. Phys. 154*, 174–196. 69, 235

Dufour, G. (2005). *Modélisation multi-fluide eulérienne pour les écoulements diphasiques à inclusions dispersées*. Ph. D. thesis, Université Paul Sabatier Toulouse III. xxiv, xxvii, 13, 21, 23, 24, 46, 50, 55, 72, 251

Dufour, G. and P. Villedieu (2005). A second-order multi-fluid model for evaporating sprays. *M2AN Math. Model. Numer. Anal. 39*(5), 931–963. xxiv, 49, 50, 65, 127, 248

Dukowicz, J. K. (1980). A particle-fluid numerical model for liquid sprays. *J. Comput. Phys. 35*(2), 229–253. xxii, 34, 35

Elghobashi, S. and G. C. Truesdell (1992). Direct numerical simulation of particle dispersion in a decaying isotropic turbulence. *J. Fluid Mech. 242*, 655–700. 35

Ern, A. and V. Giovangigli (1994). *Multicomponent transport algorithms*. Berlin: Springer-Verlag. 6, 7, 8

Faeth, G. (1983). Evaporation and combustion of spray. *Progress in Energy and Combustion Science 9*, 1–76. 21

Fevrier, P. (2000). *Etude numérique des effets de concentration préférentielle et de corrélation spatiale entre vitesses de particules solides en turbulence homogène isotrope stationnaire*. Ph. D. thesis, Institut National Polytechnique de Toulouse. xxiii, 54

Fevrier, P., O. Simonin, and K. Squires (2005). Partitioning of particle velocities in gas-solid turbulent flows into a continuous field and a spatially uncorrelated random distribution theoretical formalism and numerical study. *J.Fluid Mech 533*, 1–46. 54

Fox, R. (2007). *in "Computational Models for Turbulent Multiphase Reacting Flows"*, Udine, July 2006, Chapter Fundamental of Polydisperse Multiphase Flows. CISM Courses and Lectures. Springer Verlag. 11, 44, 52

Fox, R. O. (2008). A quadrature-based third-order moment method for dilute gas-particle flow. *Journal of Computational Physics 227*(12), 6313–6350. xxiii, xxiv, xxvi, 44, 51, 52, 110, 117, 252, 259

Fox, R. O., F. Laurent, and M. Massot (2008). Numerical simulation of spray coalescence in an eulerian framework : direct quadrature method of moments and multi-fluid method. *J. Comput. Phys. 227*(6), 2215–2222. xxiv, 52

Freret, L., S. de Chaisemartin, F. Laurent, P. Vedula, R. O. Fox, O. Thomine, J. Reveillon, and M. Massot (2009). Eulerian moment models for polydisperse weakly collisional sprays : model and validation. Proceedings of the Summer Program 2008, Center for Turbulence Research, Stanford University. xxiv, 23, 51, 55, 96, 104, 106, 118, 129, 130, 146, 252, 258, 259

Freret, L., C. Lacour, S. de Chaisemartin, F. Laurent, M. Massot, S. Ducruix, and D. Durox (2008). Pulsated free jet with polydisperse spray injection, experiments and numerical simulation. In *Proceedings of the 32nd Symp. (International) on Combustion, The Comb. Institute*, Montreal. 132

Garcia, M. (2009). *Développement et validation du formalisme Euler-Lagrange dans un solveur parallèle non-structuré pour la simulation aux grandes échelles.* Ph. D. thesis, Institut National Polytechnique de Toulouse. xxiii

Giovangigli, V. (1999a). *Multicomponent flow modeling*. Boston, MA: Birkhäuser Boston Inc. 27

Giovangigli, V. (1999b). Plane laminar flames with multicomponent transport and complex chemistry. *Math. Models Methods Appl. Sci. 9*(3), 337–378. 6, 8

Giovangigli, V. and M. Massot (1998). Asymptotic stability of equilibrium states for multicomponent reactive flows. *Math. Models Methods Appl. Sci. 8*(2), 251–297. 6

Girou, D. and G. Grasseau (1998). Optimisation mono-processeur. Technical report, French Institution IDRIS Course. 185

Glover, A., S. Skippon, and R. Boyle (1995). Interferometric laser imaging for droplet sizing: a method for droplet size measurement in sparse spray system. *Appl. Opt. 34*, 8409–8421. 135

Godlewski, E. and P.-A. Raviart (1992). *Numerical Approximation of Hyperbolic systems of conservation laws*. Applied Mathematical Sciences. New-York: Springer. 80, 85

Godsave, G. A. E. (1953). 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*, Baltimore, pp. 818–830. 19

Greenberg, J., I. Silverman, and Y. Tambour (1993). On the origin of spray sectional conservation equations. *Combustion and Flame 93*, 90–96. xxiii, xxiv, 34, 48, 49

Guermond, J.-L. and J. Shen (2001). New results on several projection methods. *C. R. Math. Acad. Sci. Paris 333*(12), 1111–1116. 217

Guichard, L., J. Reveillon, and R. Hauguel (2004). Direct numerical simulation of statistically stationary one- and two-phase turbulent combustion : a turbulent injection procedure. *Flow, turbulence and combustion 73*, 133–167. 105

Hairer, E., S. P. Nørsett, and G. Wanner (1993). *Solving ordinary differential equations. I.* Berlin: Springer-Verlag. Nonstiff problems, second revised edition. 97

Hairer, E. and G. Wanner (1996). *Solving ordinary differential equations. II.* Berlin: Springer-Verlag. Stiff and differential-algebraic problems, second revised edition. 96, 97, 98, 99, 168, 169

Hirsch, C. (1990). *Numerical Computation of Internal and External Flows: Fundamentals of Numerical Discretization, Tome II.* John Wiley & Sons. 217

Hsiang, L.-P. and G. M. Faeth (1993). Drop properties after secondary breakup. *Int. J. Multiphase Flow 19*(5), 721–735. 24

Hyland, K. E., S. McKee, and M. W. Reeks (1999). Derivation of a pdf kinetic equation for the transport of particles in turbulent flows. *J. Phys. A. 32*(34), 6169–6190. 53

Hylkema, J. (1999). *Modélisation cinétique et simulation numérique d'un brouillard dense de gouttelettes. Application aux propulseurs à poudre.* Ph. D. thesis, ENSAE. xxii, 23, 36

Hylkema, J. and P. Villedieu (1998). A random particle method to simulate coalescence phenomena in dense liquid sprays. In *Lecture Notes in Physics*, Volume 515, Arcachon, France, pp. 488–493. Proc. 16th Int. Conf. on Num. Meth. in Fluid Dyn. xxii

Ishii, M. (1975). *Thermo-Fluid Model for Two-Phase Flow.* Eyrolles. xix

Iyer, V. A. (2001). *Modeling of Diesel Sprays Using an Eulerian-Liquid Eulerian-Gas Two-Fluid Model.* Ph. D. thesis, Purdue University West Lafayette, IN. xix, xx

Jabin, P.-E. (2002). Various levels of models for aerosols. *Mathematical Models and Methods in Applied Sciences 12*(7), 903–919. 69, 75

Kah, D. (2007). Modèles eulériens pour la description de brouillards de gouttes dans des écoulements tourbillonnaires. Rapport de Master II, Ecole Centrale Paris, IFP. 50

Kah, D. (2010). *Prise en compte des aspects polydispersés dans le contexte d'une approche couplée Eulérienne-Lagrangienne pour la modélisation d'un jet de carburant dans les moteurs à combustion interne d'écoulements diphasiques.* Ph. D. thesis, Ecole Centrale Paris. 55, 106, 118, 127

Kah, D., F. Laurent, M. Massot, R. O. Fox, and S. Jay (2008). A eulerian model for the dynamics of polydisperse evaporating sprays: Combining the multi-fluid model with the quadrature method of moments. In *12^{th} SIAM International Conference on Numerical Combustion, 31/03-2/04 2008, Monteray , California.* 55

Kaufmann, A. (2004). *Vers la simulation des grandes échelles en formulation Euler-Euler des écoulements réactifs diphasiques.* Ph. D. thesis, Institut National Polytechnique de Toulouse. xxiii, 53, 54

König, G., K. Anders, and A. Frohn (1986). A new light scattering technique to measure the diameter of periodically generated moving droplets. *I. Aerosol Sci. 17*, 157–167. 135

Kuo, K. (Ed.) (1986). *Principles of combustion.* John Wiley and sons. xx

Lamarque, N. (2007). *Schémas numériques et conditions limites pour la simulation aux grandes échelles de la combustion diphasique dans les foyers d'hélicoptère.* Ph. D. thesis, Institut National Polytechnique de Toulouse. 34, 54, 85

Laurent, F. (2002a). Analyse numérique d'une méthode multi-fluide Eulérienne pour la description de sprays qui s'évaporent. *C. R. Math. Acad. Sci. Paris 334*(5), 417–422. xxiv

Laurent, F. (2002b). *Modélisation mathématique et numérique de la combustion de brouillards de gouttes polydispersés*. Ph. D. thesis, Université Claude Bernard, Lyon 1. xxiv, 13, 22, 27, 101

Laurent, F. (2006). Numerical analysis of Eulerian multi-fluid models in the context of kinetic formulations for dilute evaporating sprays. *M2AN Math. Model. Numer. Anal. 40*(3), 431–468. xxiv, 12, 49, 50, 65, 101, 107, 125, 127

Laurent, F. and M. Massot (2001). Multi-fluid modeling of laminar poly-dispersed spray flames: origin, assumptions and comparison of the sectional and sampling methods. *Comb. Theory and Modelling 5*, 537–572. xxiii, xxiv, 19, 34, 48, 49, 55, 57, 65, 110

Laurent, F., M. Massot, and P. Villedieu (2004). Eulerian multi-fluid modeling for the numerical simulation of coalescence in polydisperse dense liquid sprays. *J. Comp. Phys. 194*, 505–543. xxiv, 14, 22, 55, 65, 127

Laurent, F., V. Santoro, M. Noskov, A. Gomez, M. Smooke, and M. Massot (2004). Accurate treatment of size distribution effects in polydispersed spray diffusion flames: multi-fluid modeling, computations and experiments. *Combust. Theory and Modelling 8*, 385–412. xxiv, 27

Lecordier, B., D. Demare, L. Vervisch, J. Reveillon, and M. Trinite (2001). Estimation of the accuracy of piv treatments for turbulent flow studies by direct numerical simulation of multiphase flow. *Meas. Sci. Technol. 12*, 1382–1391. 219

Lefebre, A. H. (1989). *Atomization and Sprays*. Combustion : An International Series. Bristol: Taylor & Francis. xviii

Lele, S. K. (1992). Compact finite difference schemes with spectral like resolution. *J. Comput. Phys. 103*, 16–42. 218

LeVeque, R. J. (1992). *Numerical methods for conservation laws* (Second ed.). Basel: Birkhäuser Verlag. 80, 81, 85

LeVeque, R. J. (2002). *Finite volume methods for hyperbolic problems*. Cambridge Texts in Applied Mathematics. Cambridge: Cambridge University Press. 71, 79, 80, 81, 85

Marchisio, D. L. and R. O. Fox (2005). Solution of population balance equations using the direct quadrature method of moments. *Journal of Aerosol Science 36*, 43–73. xxiii, 44, 51, 52

Marchisio, D. L., R. D. Vigil, and R. O. Fox (2003). Quadrature method of moments for aggregation-breakage processes. *Journal of Colloid and Interface Science 258*(2), 322–334. xxiii, xxvi, 43

Marle, C. (1982). On macroscopic equations governing multiphase flow with diffusion and chemical reactions in porous media. *Int. J. Eng. Sci. 20*, 643–662. xix

Martin, C., L. Benoit, Y. Sommerer, F. Nicoud, and T. Poinsot (2006). LES and acoustic analysis of combustion instability in a staged turbulent swirled combustor. *AIAA Journal 44*(4), 741–750. xviii

Mashayek, F. (1998). Direct numerical simulations of evaporating droplet dispersion in forced low mach number turbulence. *Int. J. Heat Mass Transfer 41*, 2601–2617. xxi, 35

Mashayek, F. and R. Pandya (2003). Analytical description of particle/droplet-laden turbulent flows. *Prog. Energy Combust. Sci. 29*, 329–378. 53

Mashayek, R., F. A. Jaberi, R. S. Miller, and P. Givi (1997). Dispersion and polydispersity of droplets in stationary isotropic turbulence. *Int. J. Multiphase Flow 23*(2), 337–355. 35

Massot, M. (1996). *Modélisation Mathématique et Numérique de la Combustion des Mélanges Gazeux*. Ph. D. thesis, Thèse de l'Ecole Polytechnique. 6

Massot, M. (2007). *Eulerian Multi-fluid models for polydisperse evaporating sprays*, Chapter III of "Multi-Phase Reacting Flows: Modelling and Simulation", pp. 79–123. CISM Courses and Lectures No. 492. Springer Wien. Editors D.L. Marchisio and R. O. Fox. xxiv, 53

Massot, M., R. Knikker, C. Péra, and J. Reveillon (2004). Lagrangian/Eulerian analysis of the dispersion of evaporating sprays in non-homogeneous turbulent flows. In *International Conference on Multiphase Flows, Japan*. 54

Massot, M., M. Kumar, A. Gomez, and M. D. Smooke (1998). Counterflow spray diffusion flames of heptane: computations and experiments. In *Proceedings of the 27th Symp. (International) on Combustion, The Comb. Institute*, pp. 1975–1983. 27

Massot, M., F. Laurent, D. Kah, and S. de Chaisemartin (2009). A robust moment method for evaluation of the disappearance rate of evaporating sprays. *SIAM J. Appl. Math.*. submitted for publication. xxiv, 50, 52, 54, 127, 145, 259

Massot, M. and P. Villedieu (2001). Modélisation multi-fluide eulérienne pour la simulation de brouillards denses polydispersés. *C. R. Acad. Sci. Paris Sér. I Math. 332*(9), 869–874. 55

McGraw, R. (1997). Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology 27*, 255–265. xxiii, 43, 44

Meftah, H. (2008). *Simulation numérique directe d'un spray en évaporation : Analyse et modélisation du mélange turbulent et des transferts thermiques*. Ph. D. thesis, INSA de Rouen. 105, 248, 252

Menard, T., S. Tanguy, and A. Berlemont (2007). 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. xx, xxi

Metcalf, M., J. Reid, and M. Cohen (2004). *fortran 95/2003 explained* (first ed.). Numerical Mathematics and Scientific computation. Oxford: Oxford University Press. 148, 151

Miller, R. S. and J. Bellan (1999). Direct numerical simulation of a confined three-dimensional gas mixing layer with one evaporating hydrocarbon-droplet-laden stream. *Journal of Fluids Mechanics 384*, 293–338. 34, 35

Miller, R. S. and J. Bellan (2000). Direct numerical simulation and subgrid analysis of a transitional droplet laden mixing layer. *Phys. Fluid 12*(3), 650–671. xxi, 35

Minier, J. and E. Peirano (2001). The pdf approach to turbulent polydispersed two-phase flows. *Physics Reports 352*, 1–214. 53

Moreau, M. (2006). *Modélisation numérique directe et des grandes échelles des écoulements turbulents gaz-particules dans le formalisme eulérien mésoscopique*. Ph. D. thesis, Institut National Polytechnique de Toulouse. 54

Mossa, J.-B. (2005). *Extension polydisperse pour la description euler-euler des écoulements diphasiques réactifs - TH/CFD/05/74*. Ph. D. thesis, Institut National Polytechnique de Toulouse. 47, 54

Murrone, A. and H. Guillard (2005). A five equation reduced model for compressible two phase flow problems. *J. Comp. Phys. 205*, 664–698. 44

Nicoud, F. (2000). Conservative high-order finite-difference schemes for low-mach number flows. *J. Comp. Phys. 158*, 71–97. 217

Noiray, N., D. Durox, T. Schuller, and S. Candel (2008). A unified framework for nonlinear combustion instability analysis based on the flame describing function. *Journal of Fluid Mechanics 615*, 139–167. xvii

O'Rourke, P. J. (1981). *Collective drop effects on vaporizing liquid sprays*. Ph. D. thesis, Princeton University. xx, xxii, xxvi, 6, 15, 34, 35, 40

O'Rourke, P. J. and A. Amsden (1987). The tab method for numerical calculation of spray droplet breakup. Technical Report 87545, Los Alamos National Laboratory, Los Alamos, New Mexico. 24

Overholt, M. R. and S. B. Pope (1998). A deterministic forcing scheme for direct numerical simulation of turbulence. *Computer and Fluid 27*, 11–28. 105

Péra, C. (2005). *Simulation et modélisation de la dispersion turbulente, l'évaporation et la combustion d'écoulements diphasiques*. Ph. D. thesis, INSA de Rouen. 105, 219

Pichard, C., D. Durox, and S. Ducruix (2005). Effects of acoustic modulations on a spray flame. In *20th ILASS, Annual Meeting of the Institute of Liquid Atomization and Spray System*, Orléans, France. 133, 134

Poinsot, T. and S. K. Lele (1992). Boundary conditions for direct simulations of compressible viscous flows. *J. Comput. Phys. 101*(1), 104–129. 218

Poinsot, T. and D. Veynante (2005). *Theoretical and Numerical Combustion* (second ed.). Philadelphia: Edwards. xvii, xviii, 6, 7, 8, 9, 30, 31

Pope, S. (2000). *Turbulent flows*. Cambridge University. xvii

Popinet, S. and S. Zaleski (1999). A front-tracking algorithm for accurate representation of surface tension. *Int. J. Numer. Meth. Fluids 30*, 775–793. xxi

Ramkrishna, D. and A. G. Fredrickson (2000). *Population Balances: Theory and Applications to Particulate Systems in Engineering*. Academic Press. xxiii, 43

Reeks, M. (1992). On the continuum equations for dispersed particles in nonuniform flows. *Phys. Fluids A 4*(6), 1290–1303. 53

Reeks, M. (1993). On the constitutive relations for dispersed particles in nonuniform flows. I: Dispersion in simple shear flow. *Phys. Fluids A 5*(3), 750–761. 53

Reeks, M. W. (1991). On a kinetic equation for the transport of particles in turbulent flows. *Phys. Fluids 3*, 446–456. xxiii, 53

Reveillon, J. (2007). *in "Computational Models for Turbulent Multiphase Reacting Flows"*, Udine, July 2006, Chapter DNS of Spray Combustion, Dispersion Evaporation and Combustion. CISM Courses and Lectures. Springer Verlag. 17, 18, 19, 20

Reveillon, J., N. Bray, and L. Vervish (1998). DNS study of spray vaporization and turbulent micro-mixing. In *AIAA 98-1028, 36th Aerospace Sciences Meeting and Exhibit*, Reno NV. 34, 35

Reveillon, J. and F. Demoulin (2007). Effects of the preferential segregation of droplets on evaporation and turbulent mixing. *J. Fluid Mech. 583*, 273–302. 220, 248, 249, 250, 252

Reveillon, J., C. Péra, and M. Massot (2002). Lagrangian/Eulerian analysis of the dispersion of vaporizing polydispersed sprays in turbulent flows. Proceedings of the Summer Program 2008, Center for Turbulence Research, Stanford University. 54

Reveillon, J., C. Péra, M. Massot, and R. Knikker (2004). Eulerian analysis of the dispersion of evaporating polydispersed sprays in a statistically stationary turbulent flow. *Journal of Turbulence 5*(1), 1–27. 54, 259

Reveillon, J. and L. Vervisch (2000). Spray vaporization in nonpremixed turbulent combustion modeling: a single droplet model. *Combustion and Flame 121*, 75–90. 219

Reveillon, J. and L. Vervisch (2005). Analysis of weakly turbulent diluted-spray flames and spray combustion regimes. *J. Fluid Mech. 537*, 317–347. xix, xxi, 35

Rey, C. (2004). *Interactions collectives dans les instabilités de combustion à haute fréquence*. Ph. D. thesis, Ecole Centrale Paris. xviii

Riber, E. (2007). *Développement de la méthode de simulation aux grandes échelles pour les écoulements diphasiques turbulents*. Ph. D. thesis, Institut National Polytechnique de Toulouse. 54

Riber, E., M. Garcia, V. Moureau, H. Pitsch, O. Simonin, and T. Poinsot (2006). Evaluation of numerical strategies for les of two-phase reacting flows. Proceedings of the Summer Program 2006, Center for Turbulence Research, Stanford University. 54

Richard, S. (2005). *Simulation aux grandes échelles de la combustion dans les moteurs à allumage commandé*. Ph. D. thesis, Ecole Centrale Paris. xviii

Riley, J. J. and G. S. Patterson (1974). Diffusion experiments with numerically intergrated isotropic turbulence. *Physics of Fluids 17*, 292–297. 35

Rüger, M., S. Hohmann, M. Sommerfeld, and G. Kohnen (2000). Euler/Lagrange calculations of turbulent sprays: the effect of droplet collisions and coalescence. *Atomization and Sprays 10*(1), 47–81. 34

Schiller, L. and A. Naumann (1935). A drag coefficient correlation. *V.D.I Zeitung 77*, 318–320. 15

Schmitt, P., T. Poinsot, B. Schuermans, and K. Geigle (2007). Large-eddy simulation and experimental study of heat transfer, nitric oxide emissions and combustion instability in a swirled turbulent high pressure burner. *J. Fluid Mech. 570*, 17–46. xvii

Selle, L., L. Benoit, T. Poinsot, F. Nicoud, and W. Krebs (2006). Joint use of compressible large-eddy simulation and helmoltz solvers for the analysis of rotating modes in an industrial swirled burner. *Combustion and Flame 145*(1-2), 194–205. xviii

Sinkovits, R. (2001). Single processor optimization. In *Parallel computing class, University of Sydney, April-May 2001*. 185

Sirignano, W. A. (1999). *Fluid Dynamics and Transport of Droplets and Sprays*. Cambridge: Cambridge University Press. 13, 14, 15, 17, 19, 20, 35

Spalding, D. B. (1953). The combustion of liquid fuels. In *Proceedings of the 4th Symp. (International) on Combustion, The Comb. Institute*, Baltimore, pp. 847–864. 19

Squires, K. D. and J. K. Eaton (1991). Preferential concentration of particles by turbulence. *Physics of Fluids 3*, 1169–1178. 35

Strang, G. (1968). On the construction and comparison of difference schemes. *SIAM J. Num. Anal. 5*, 507–517. 78

Subramanian, S. (2000). Statistical representation of a spray as a point process. *Physics of Fluids 12*. 220

Temam, R. (1968). Une méthode d'approximation de la solution des équations de Navier-Stokes. *Bull. Soc. Math. France 96*, 115–152. 217

Truchot, B. (2005). *Développement et validation d'un modèle eulérien en vue de la simulation des jets de carburants dans les moteurs a combustion*. Ph. D. thesis, Institut National Polytechnique de Toulouse. xix, xx

van Leer, B. (1977). Towards the ultimate conservative difference scheme IV. a new approach to numerical convection. *J. Comput. Phys. 23*, 276–299. 82

Vedula, P., R. O. Fox, and I. Boyd (2008). A quadrature-based method of moments for solution of the collisional boltzmann equation. *Journal of Statistical Physics*. Submitted. 22

Versaevel, P. (1996). *Combustion laminaire diphasique: Etude théorique et expérimentale*. Ph. D. thesis, Ecole Centrale Paris. 16

Vervisch, L., B. Labegorre, and J. Reveillon (2004). Hydrogen-sulphur oxy-flame analysis and single-step flame tabulated chemistry. *Fuel 83*, 605–614. 30

Vessiller, C. (2007). *Contribution à l'étude des brouillards denses et dilués par la simulation numérique Euler/Euler et Euler/Lagrange.* Ph. D. thesis, Ecole Centrale Paris. xx

Villedieu, P. and J. Hylkema (1997). Une méthode particulaire aléatoire reposant sur une équation cinétique pour la simulation numérique des sprays denses de gouttelettes liquides. *C. R. Acad. Sci. Paris Sér. I Math. 325*(3), 323–328. 14, 21

Villedieu, P. and J. Hylkema (2000). Modèles numériques lagrangiens pour la phase dispersée dans les propulseurs à poudre. Rapport Final 1/3784.00/DTIM, ONERA. 23

Volkov, A., Y. Tsirkunov, and B. Oesterle (2005). Numerical simulation of a supersonic gas-solid flow over a blunt body : The role of inter-particle collisions and two-way coupling effects. *Int. J. Multiphase Flow 31*(12), 1244–1275. 72

Wang, L. P. and M. R. Maxey (1993). Settling velocity and concentration distribution of heavy particles in homogeneous isotropic turbulence. *J. Fluid Mech. 256*, 27–68. 249

Wert, K. L. (1995). A rationally-based correlation of mean fragment size for drop secondary breakup. *Int. J. Multiphase Flow 21*(6), 1063–1071. 24

Williams, F. A. (1958). Spray combustion and atomization. *Phys. Fluids 1*, 541–545. 11, 12

Williams, F. A. (1985). *Combustion Theory (Combustion Science and Engineering Series)*. ed F A Williams (Reading, MA: Addison-Wesley). 11

Wray, A. A. (1990). Minimal storage time-advancement schemes for spectral methods. Technical report, Center for Turbulence Research, Stanford University. 218

Zel′dovich, Y. B. (1970). Gravitational instability : an approximate theory for large density perturbations. *Astronomy and Astrophysics 5*, 84–89. 69