Modélisation et simulation Eulériennes des écoulements diphasiques à phases séparées et dispersées : développement d'une modélisation unifiée et de méthodes numériques adaptées au calcul massivement parallèle

Florence Drui

NNT : 2017SACLC033

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

Ecole doctorale n°574
Ecole doctorale de mathématiques Hadamard
Spécialité de doctorat: Mathématiques aux interfaces

par

Mme. Florence Drui

Modélisation et simulation Eulériennes des écoulements diphasiques à phases séparées et dispersées : développement d'une modélisation unifiée et de méthodes numériques adaptées au calcul massivement parallèle

Thèse présentée et soutenue à CentraleSupelec - campus de Chatenay-Malabry, le 07 juillet 2017.

Composition du Jury :

M.	Philippe Helluy	Professeur Université de Strasbourg	(Président du jury)
M.	Carsten Burstedde	Professeur Rheinische Friedrich-Whilhelms-Universität, Bonn	(Rapporteur)
M.	Sergey Gavrilyuk	Professeur Aix-Marseille Université	(Rapporteur)
M.	Christophe Benoit	Ingénieur ONERA, PAlaiseau	(Examinateur)
M.	Frédéric Coquel	Directeur de recherche Ecole Polytechnique, Palaiseau	(Examinateur)
M.	Adam Larat	Chargé de recherche CNRS	(Encadrant)
M.	Samuel Kokh	Ingénieur-Chercheur CEA-Direction de l'Energie Nucléaire	(Co-directeur de these)
M.	Marc Massot	Professeur CentraleSupelec, Ecole Polytechnique	(Directeur de these)

Remerciements

Je souhaite remercier les différents organismes qui ont participé au financement de ces travaux : le CEA et la DGA, CentraleSupélec et le CNRS à travers l'ANR Sub Super Jet (ANR-14-CE22-0014), ainsi que GENCI-CINES pour l'accès aux ressources de calcul.

Je profite également de cette page pour remercier l'ensemble des acteurs, plus ou moins impliqués, dans ce travail de thèse. Et tout d'abord, rien ne serait sans mes trois encadrants et co-directeurs de thèse. Merci à Adam pour cette collaboration de plus de quatre ans, depuis mon projet de troisième année à Centrale, pour toutes les séances de travail, de code et débuggage, de répétitions de soutenance ! Le travail présenté dans cette thèse trouve un sens lorsque l'on regarde l'ensemble des travaux de Marc sur la modélisation et la simulation des écoulements diphasiques. Merci à toi de m'avoir pris en thèse et d'avoir partagé tes connaissances et ta vision scientifique qui a toujours un temps d'avance ! Enfin, Samuel, j'ai appris énormément avec toi, des points de vue scientifique, méthodologique et humain, à travers ta gentillesse à toute épreuve !

Merci beaucoup aussi à Vincent, qui a été présent pendant les premiers mois de ma thèse et avec qui j'espère bien avoir à collaborer à nouveau, et à Pierre de la Maison de la Simulation, sans qui il n'y aurait pas CanoP, et qui a eu la patience de relire certains chapitres du manuscrit. Enfin, j'ai une pensée pour tous les membres de l'équipe de mathématiques, à l'EM2C ou à la MdlS. Mohamed pour tout ce qu'on a fait ces deux dernières années, je te souhaite le meilleur pour la fin de thèse et pour la suite. Alex je garde des très bons souvenirs de l'été 2014 et en particulier du CEMRACS. Aymeric, Frédérique, Stéphane, Francois, Valentin, Macole, Pierre, Marc-Arthur, David et maintenant aussi Ruben pour le côté EM2C/CMAP et Maxime du côté MdlS, j'ai été très contente d'avoir pu échanger et travailler avec vous.

Un grand merci aussi aux membres du jury, qui ont accepté de lire le manuscrit et qui ont fait le déplacement pour la soutenance : Sergey pour les discussions et échanges au cours de divers séminaires, Carsten pour l'intérêt que vous avez porté à mes travaux, Frédéric pour avoir pris le temps de lire ce pavé et avoir été présent lors de la soutenance, Philippe pour avoir gardé vos questions pour plus tard, Donna pour votre rapport et nos échanges par e-mail, merci aussi à Christophe et Lucien.

Et puis la thèse, c'est aussi trois années dans les laboratoires et en-dehors ! Merci aussi à tous mes collègues de Centrale et de la Maison de la Simulation et en particulier à ceux qui participent à la vie sociale de ces deux laboratoires. J'ai passé avec vous des très agréables moments, que ce soit à discuter autour d'un verre avec Julien, Fabien, Maxime, Rehan, Thibault, Pascal, Sophie, Matthieu, Nathalie, Ralitsa pour la MdlS ou Clément, Aurélien, Fabien pour EM2C, ou simplement au bureau avec Thibault, Renaud, Maria, Yi, Jeaniffer. Il y a eu également les séances de sport du vendredi midi avec Marien, Adrien, Laurent, Paul, Manuel, Nicolas, Yunhui, Aude, Chai, Pierre, Jérôme, Saufi, Jérémie, Artur, Gilles (et j'en oublie malheureusement peut-être)...

Enfin, parce que vous avez toujours été là pour me soutenir, merci à ma famille et mes amis, et parce que j'ai la chance de pouvoir partager ma vie et certains problèmes de maths avec toi, merci Gilles !

Résumé

Dans un contexte industriel, l'utilisation de modèles diphasiques d'ordre réduit est nécessaire pour pouvoir effectuer des simulations numériques prédictives de l'injection de combustible liquide dans les chambres de combustion automobiles et aéronautiques. En effet, les résultats de ces simulations servent à la conception de nouveaux équipements, plus performants, et qui permettent de réduire la consommation de carburant ainsi que l'émission de polluants. L'un des facteurs clés d'une combustion de "bonne qualité" est l'amélioration du processus d'atomisation lors de l'injection : la taille et la répartition spatiale des gouttelettes créées lors de l'atomisation sont en effet déterminants lors de la combustion. Cependant, ce processus d'injection, depuis le régime diphasique de phases séparées à la sortie de l'injecteur, jusqu'au régime de phase dispersée qui caractérise le brouillard de gouttelettes, est un problème impliquant des mécanismes physiques dans une très large gamme d'échelles. Aujourd'hui, la dérivation d'un modèle d'ordre réduit capable de prendre en compte toutes ces échelles, mais aussi le développement de méthodes numériques associées qui soient précises et robustes, demandent des avancées majeures en modélisation, méthodes numériques et calcul haute performance (HPC). Ces trois aspects sont abordés dans cette thèse.

La première partie de ce travail est donc consacrée au développement de modèles diphasiques, permettant de décrire plusieurs régimes et configurations d'écoulement. Parmi les approches proposées dans la littérature, les modèles de mélange compressibles semblent être un point de départ intéressant, puisqu'ils sont couramment utilisés dans des méthodes de capture d'interface pour les écoulements à phases séparées. Ces méthodes ne nécessitent pas de reconstruction de l'interface, celle-ci est située dans une zone de diffusion liée choix de discrétisation des équations. A partir des modèles de mélange, et de la même manière que pour la dérivation de modèles bifluides décrivant les milieux à bulles, le principe variationnel de Hamilton et le second principe de la thermodynamique sont utilisés ici pour enrichir ces modèles et décrire des effets de sous-échelles, choisis a priori. Ces effets correspondent à de la pulsation de l'interface et sont pilotés par deux paramètres: une micro-inertie et une micro-viscosité. Une hiérarchie de modèles reliés par ces deux paramètres est alors dérivée : le modèles le plus complet de cette hiérarchie (modèle à 5 équations) intègre ces effets de micro-pulsation, tandis que le modèle le plus réduit de la hiérarchie (modèle à 3 équations) correspond au modèle de mélange des méthodes de capture d'interface. Le modèle à 5 équations est comparé à un modèle de référence et à des données expérimentales dans un contexte de milieux à bulles. Cette comparaison permet de vérifier la pertinence de ce modèle pour décrire les effets de sous-échelle et aussi d'identifier analytiquement les expressions de la micro-inertie et de la micro-viscosité. Enfin, une étude de la description géométrique et statistique des interfaces permet d'envisager de nouvelles pistes pour permettre aux modèles de mélange diphasiques une description d'effets interfaciaux plus généraux et ainsi d'être couplés avec les modèles cinétiques pour la description de la phase dispersée.

La seconde partie des travaux présentés dans cette thèse concerne une stratégie de discrétisation des équations adaptée aux modèles de mélange diphasiques compressibles. D'une part, le caractère bien posé de ces modèles permet d'adapter des solveurs de Riemann approchés classiques et de résoudre la partie convective des équations. Les adaptations de ces schémas permet en outre de remédier aux difficultés liées au rapport élevé des impédances acoustiques et des densités des deux fluides et qui, pour certains schémas génèrent des surcoûts de calcul. Grâce à une technique de séparation d'opérateurs, les termes sources sont intégrés indépendamment, à l'aide de solveurs d'équations différentielles ordinaires appropriés et gérant la raideur des termes sources. Enfin, la nature compressible des équations est connue pour être mal résolue par des schémas de type Godunov dans des régimes d'écoulement à faibles nombres de Mach. A partir de

travaux de la littérature, nous proposons des schémas numériques qui résolvent ces difficultés. D'autre part, nous proposons d'adapter certaines méthodes de volume finis, généralement utilisées sur des grilles structurées régulières, à des grilles non conformes, caractéristiques de grilles adaptées dynamiquement (maillages AMR). L'objectif des méthodes présentées est d'augmenter la précision de la solution dans les zones proches de l'interface, pour limiter sa diffusion. Ces méthodes sont testées sur des problèmes de Riemann classiques, des tests 1D et des simulations d'écoulements interfaciaux. En particulier, des simulations de rupture de barrage permettent de tester la stratégie numérique dans des configurations d'écoulements bas-Mach. Les comparaisons avec des données expérimentales montrent que malgré la simplicité du modèle, la dynamique de l'écoulement liquide est très bien reproduite par la simulation.

Enfin, la dernière partie aborde le développement d'un code de calcul AMR, ayant une bonne scalabilité sur des architectures massivement parallèles et permettant la simulation de modèles diphasiques. Basé sur la bibliothèque AMR p4est, le code CanoP a été développé conjointement par la Maison de la Simulation, le laboratoire EM2C de CentraleSupélec et l'IFPEn. CanoP permet d'utiliser des schémas de volumes finis pour résoudre des équations de la mécanique des fluides, telles que les modèles diphasiques, des modèles de spray ou les équations d'Euler. Des résultats de performance sur la machine Occigen, ainsi que des simulations 2D et 3D sont ici réalisés et analysés.

L'ensemble du travail constitue une base pour des études plus approfondies des écoulements diphasiques et des développements de modèles physiques, vers une modélisation unifiée, ainsi que des méthodes numériques vers des méthodes d'ordres plus élevés. Grâce à sa modularité et à sa scalabilité parallèle, le code CanoP permet de tester ces nouveaux modèles et schémas pour des coûts de développement et de simulation réduits.

Contents

List of Figures

Introduction

0.1 General context

0.1.1 Multiphase flows in environmental and industrial contexts

Both our present daily life and long term future are determined by the access and use of energy. That is the reason why an efficient use of available resources raises multiple challenges, like enabling savings on their use. This is especially true when these resources are non-renewable. Moreover, the production of energy often generates pollutants. Controlling and reducing the emissions of these pollutants represent today the second greatest challenge of the energy sector. Nowadays, fossil fuels still represent a major source of energy, and more particularly in the transport sector. Due to their increasing scarcity and increasing cost of extraction, combustion devices relying on this source must be designed so as to improve the efficiency of the burning processes. A combustion of *better quality* implies also lower productions of CO_2, which is one of the current main greenhouse gases (in terms of mass it is the most widely produced greenhouse gas), of nitrogen oxides (NO_x), of soots and other particles, which may pose a health risk. Many research studies are conducted in order to better understand the combustion mechanisms and to develop more efficient engines. Among the critical mechanisms involved in fuel combustion in aeronautical and automotive engines, multiphase flows play a key role.

Multiphase flows are characterized by the presence of two or more phases with different properties. For example, one phase may be liquid while the other is gaseous or solid. An interface delimits the zones of validity of their intrinsic properties. This interface evolves according to the material velocities of the phases, to phase transition or to interfacial energies. Multiphase flows can take a large variety of forms and are encountered in a wide range of natural and industrial processes: their modeling is thus at the heart of numerous studies. One can mention their presence in numerous energy devices, such as thermal power plants, where energy is transferred through phase changes, or in nuclear plants, where they are studied for security purposes. Fluidized beds involve solid-gas or solid-liquid two-phase flows and are used in the chemical and food industries (Tsuji, Kawaguchi, and Tanaka (1993)). Understanding the interactions between two-phase flows and mechanical structures is also of large interest in coastal engineering or in the development of marine energy devices: for example, wave breaking can have damaging effects on these structures, whose design must then resist these effects (Hayatdavoodi and Ertekin (2016)). Natural multiphase flows are also the subjects of multiple research studies, such as the motions of particles in the atmosphere or of sediments in the oceans. One can also find studies on the prevention of natural hazards, such as avalanches or forest fires. Finally, in the astrophysical domain, the formation of planets or protoplanetary disks can be described by means of multiphase flows models (Cuzzi, Hogan, and Shariff (2008)).

The objective of most of these studies is to be able to predict the evolution of multiphase flows in a given configuration. These predictions may be achieved by using models, *i.e.* sets of equations, whose solutions give more or less accurate descriptions of reality. Developing new models or improving existing ones may be done in several ways. For example, complex phenomena can be better understood from observations and analysis of experimental studies. Experimental data are also used to validate models predictions. Models can be developed from theoretical and mathematical grounds. Then, they may predict physical mechanisms that have not yet been observed experimentally. A third approach consists in using the numerical simulation to better understand some mechanisms that are experimentally hardly observable.

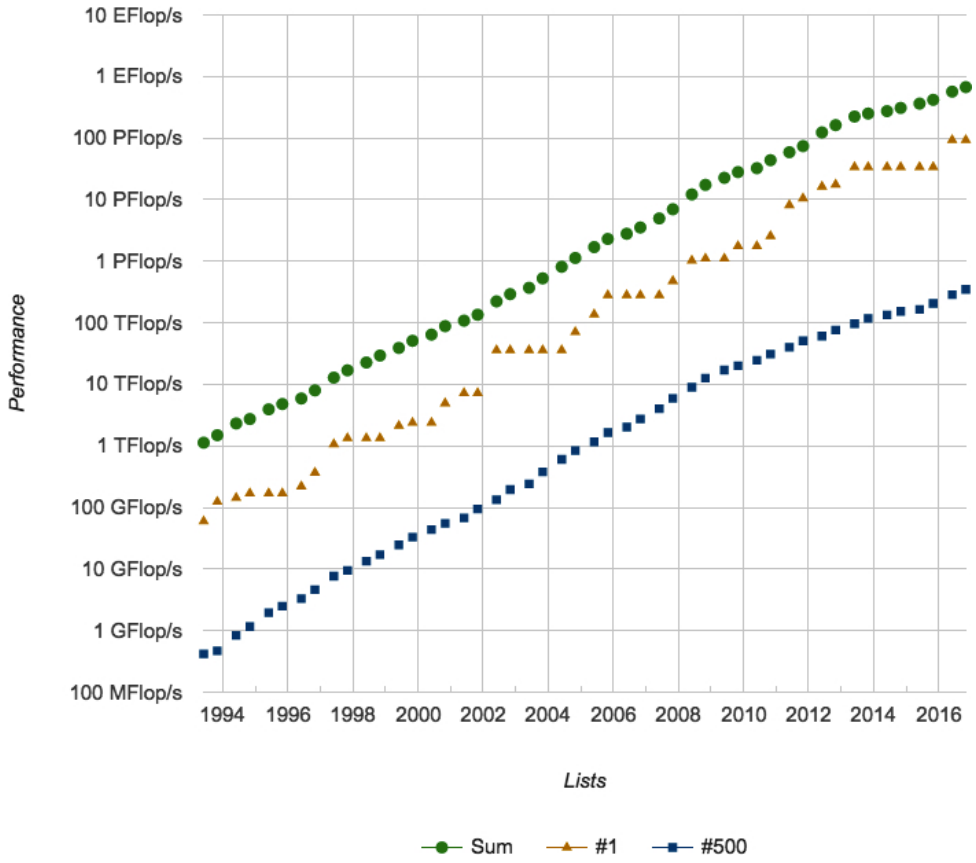


Figure 1: *Exponential growth of supercomputing power as recorded by the TOP500 list. From TOP500 (2017)*

Numerical simulations however necessitate a model, which is solved with appropriate numerical tools. One category of numerical simulations is called **Direct Numerical Simulation** (DNS): for a given model and a given physical phenomenon, a range of scales may be determined for the validity of the model. Then, the resolution of all the scales of the problem is ensured by a discretization of the equations that is as fine as the finest physical scales. For turbulent monophasic flows for example, the finest scale is the Kolmogorov scale. However, for two-phase flows and more specifically for the resolution of the interfacial phenomena, the smallest scales are often hard to determine. We refer to Scardovelli and Zaleski (1999) and related works for examples of DNS attempts of two-phase flows. DNS can also be used to derive averaged laws for a given system, such as experiments do. Indeed, the study of the whole system may help understand *macroscopic*-scale or collective effects. However, DNS requires considerable computational resources. The increasing development of the High Performance Computing (HPC) infrastructures, has facilitated DNS use. This trend is illustrated in Figure 1, that shows the evolution during these last years of the performance of the whole HPC resources, of the first and the 500^{th} machines worldwide. Hence, ever bigger and more complex problems can be simulated with DNS (Chen (2011)). But numerous local smaller clusters also allow numerical simulations to develop, in the academic as well as in the industrial sectors. With the computational resources available on such machines, DNS is often not tractable. Other approaches must be developed: since one cannot simulate all the scales of the physical problems, one must consider models, which are valid at larger physical scales. These models are called **reduced-order** models. Moreover, the numerical tools used to solve these models must be reliable, accurate and performant, so that physically relevant results can be obtained for a reasonable computational cost.

Simulations of combustion in automotive (and aeronautical) engines are examples of complex problems, for which DNS may be very expensive. However, simulation tools are essential for the design of new mechanical devices, or to study the origins of the pollutant emissions. Among the main focus is the

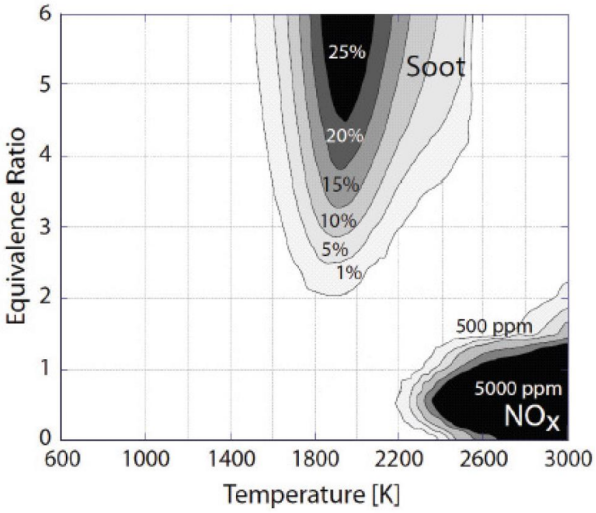


Figure 2: *Pollutant formation in a ϕ - T diagram. Source: Kamimoto and hoan Bae (1988)*

simulation of the combustion mechanisms, for the reasons that are mentioned in the next section. Numerical strategies for this simulation must be affordable in an industrial context, where both time and computational resources are valuable.

0.1.2 Overview of current issues about combustion and combustion engines

Let us give an overview of the mechanisms which may reduce the combustion efficiency and cause the formation of pollutants in combustion engines. More specifically, let us have a look at the two main categories of reciprocating engines used in the automotive sector: the gasoline and Diesel engines. For both of them, a mechanical energy is produced by the expansion of burnt gases, issued from the combustion of a fuel with an oxidizer (the air).

Usually, in gasoline engines, air and gasoline are premixed before entering the combustion chamber. Combustion is then activated by spark ignition. A certain quality of the combustion is ensured by the good mixing of air and fuel and the homogeneity of the air-fuel ratio, close to the stoichiometric mixture in the combustion chamber.

Diesel engines are based on the injection of the liquid fuel directly into the chamber and ignition occurs due to the air compression. In a first stage of the combustion process, the fuel that has been injected before ignition starts burning. In a second stage, the flame is located near the fuel jet. Finally, when the fuel injection is over, the remaining unburnt fuel may continue to burn, according to its location and the conditions inside the chamber. The pollutants are mainly produced during this last stage, where the combustion is less controlled and suitable conditions for their development can be found. Figure 2 shows a diagram of conditions for soots and nitrogen oxides formations. A part of the current research fields about combustion aims thus at understanding precisely the combustion mechanisms and at the modeling of the mechanical and thermodynamical processes, related to the pollutants formations. Furthermore, it is essential to ensure a combustion, as good as possible, during the stage of injection. During this critical stage, the fuel, injected at a high-pressure and high-velocity regime, is atomized and creates a cloud of very small droplets. Due to the pressure and temperature conditions in the chamber, these droplets are then partially vaporized, before this air-fuel mixture starts to burn. If the droplets are too large, they will not fully evaporate and will not burn during the injection stage. Understanding and improving the atomization process, so as to get droplets as small as possible, is a research topic that has been studied by many authors over the years.

In the following sections, we first present the main physical mechanisms involved in the atomization process, and the physical scales that are associated with these mechanisms. This description will show that atomization is a multi-scale process, whose modeling and simulation represent still challenges. We

will present the different approaches that can be found in the literature to deal with this problem, and show their limits. Then, we will focus on computational techniques used in a HPC context. Finally, we will present the issues, about modeling and discretization strategies, addressed in the present work, and our contributions to tackle these issues.

0.2 Physical description of the atomization processes

As we have mentioned in the first section, one key parameter for a combustion of good quality lies in the way the liquid fuel is atomized. Let us now give a first description of this process. This will highlight the possible modeling issues.

0.2.1 Primary and secondary break-ups

In Figure 3 we present a sketch of the liquid fuel injected in the chamber. Initially in a one piece of space volume, called the liquid core, the liquid phase going out of the injector undergoes instabilities that deform the gas-liquid interface. These deformations grow and create ligaments. Then, due to different break-up mechanisms, these ligaments are split into drops that are detached from the main liquid core: this is the **primary break-up**. Due to interface deformations, bubbles may enter the liquid phase. Bubbles may also be created by cavitation phenomena, because of strong pressure variations. These phenomena may appear within the nozzle (Reitz and Bracco (1982)). Finally, the drops that have detached from the main liquid core can be divided again, creating then a cloud of fine droplets: this is the **secondary break-up**.

Understanding precisely the break-up phenomena is still an open research area, and only recent experiments (Marmottant and Villermaux (2004)) or DNS results (Berlemont, Ménard, Aniszewski, and Vaudor (2016)), have shed some light on this matter. However, it clearly appears that the atomization is the result of instabilities that grow at the interface between the two fluids. Different phenomena are competing, among which, the relative inertia of the two phases, their velocities, their viscosities and surface tension. A classic approach in fluid mechanics consists in analyzing dimensionless numbers that characterize the flow.

0.2.2 Surface instabilities and dimensionless numbers

Instabilities of different natures contribute to a more or less large extent to the atomization process (see an illustration in Figure 4). These instabilities are:

- the Kelvin-Helmholtz instabilities, defined by the interface deformation due to a velocity differential between the two fluids, and the effects of viscosity on the transitional velocity profile,
- the Plateau-Rayleigh instability, occurring at low Reynolds (no turbulence) and low Weber (predominance of surface tension effects) numbers, after the interface has undergone Kelvin-Helmholtz instabilities. When these deformations are large enough, the surface forces lead to the separation of drops from the main liquid core,
- with increasing Weber numbers, the Kelvin-Helmholtz instabilities occur closer to the nozzle exit.

In general, increasing the velocity differential between the liquid and the gas phase tends to increase the break-up effects. Indeed, increasing the velocity generates more turbulence and also increases the aerodynamic forces (see Figure 5). The prevalence of these instabilities according to the flow conditions can be characterized by the following dimensionless numbers:

- the Reynolds number, Re, characterizes the ratio of the inertial forces with viscous ones and the turbulent regime of the flow (index l is for the liquid phase, g for the gas phase):

$$\mathrm{Re}_{l,g} = \frac{\rho_{l,g} u_{l,g} D_{l,g}}{\mu_{l,g}}.$$

  Turbulence within the liquid phase contributes to its surface deformation,

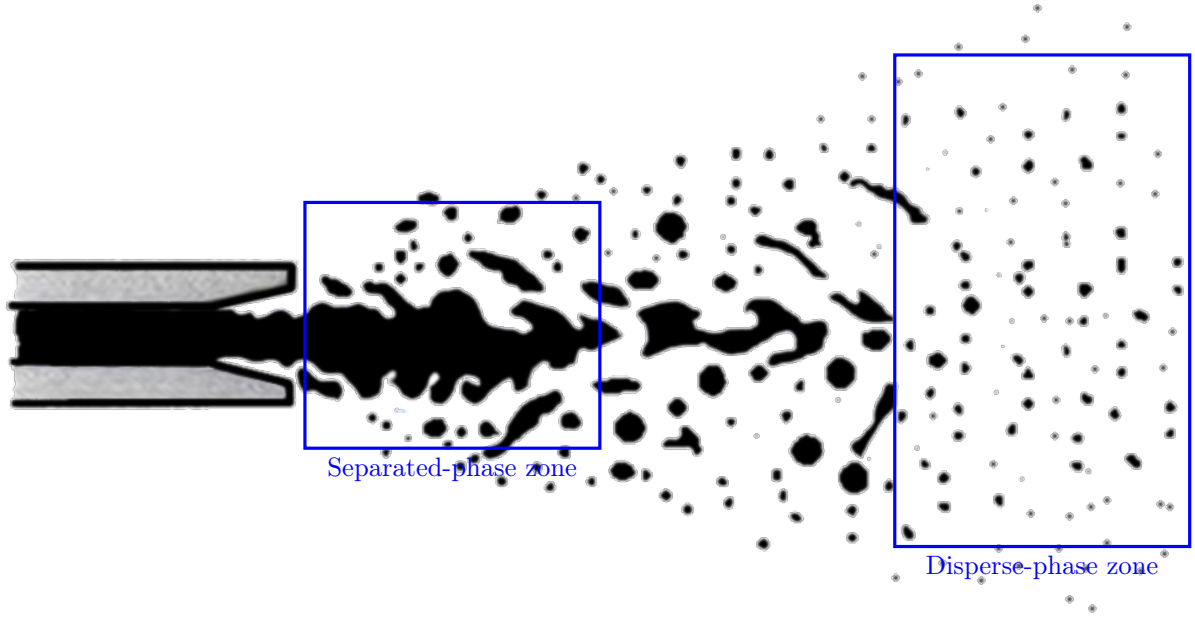


Figure 3: *Sketch of a liquid fuel injection and examples of separated-phases and disperse-phase zones.*

- the Weber number We characterizes the ratio of the inertial forces with the surface tension effects. It may be considered either as an external force ratio applied to the liquid volume by the gas motion:

$$\mathrm{We} = \frac{\rho_g (u_g - u_l)^2 D_l}{\sigma_l},$$

or from the liquid side:

$$\mathrm{We}_l = \frac{\rho_l u_l^2 D_l}{\sigma_l}.$$

The velocity differential between the two phases contributes to interfacial instabilities, while surface tension maintains some regularity of the interface (Figure 6),

- the ratio of kinetic energies J is used in cases where the gas motion helps the atomization of the liquid core (air-assisted atomization, cross-flow configuration):

$$\mathrm{J} = \frac{\rho_g u_g^2}{\rho_l u_l^2},$$

- for the ratio of viscosity forces with surface tension effects, one usually considers the Ohnesorge number Oh defined for the liquid phase by:

$$\mathrm{Oh} = \frac{\sqrt{\mathrm{We}_l}}{\mathrm{Re}_l} = \frac{\mu_l}{\sqrt{\rho_l \sigma D_l}}.$$

0.2.3 Parameters of the atomization in Diesel engines

Let us complete this part dedicated to the physical description of the atomization processes by giving some orders of magnitudes of the injection parameters and dimensionless numbers in Diesel engines

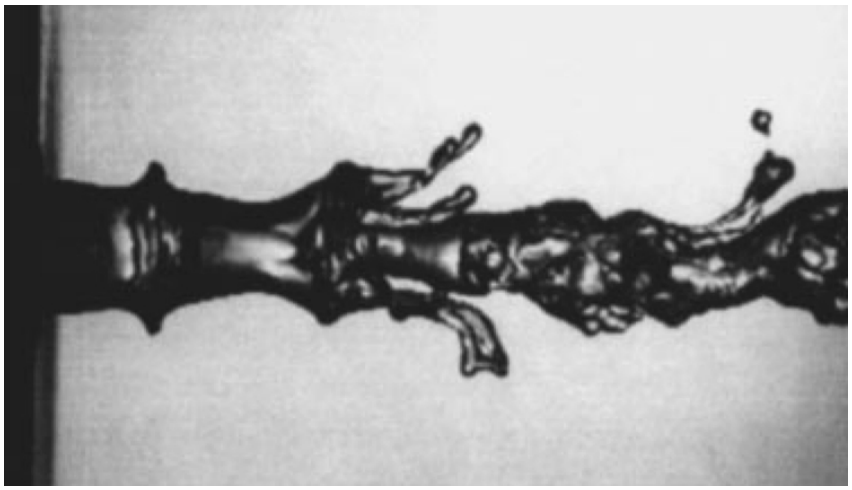


Figure 4: *Photograph of the instabilities that can develop at the surface of the main liquid core (picture from Lasheras and Hopfinger (2000)).*

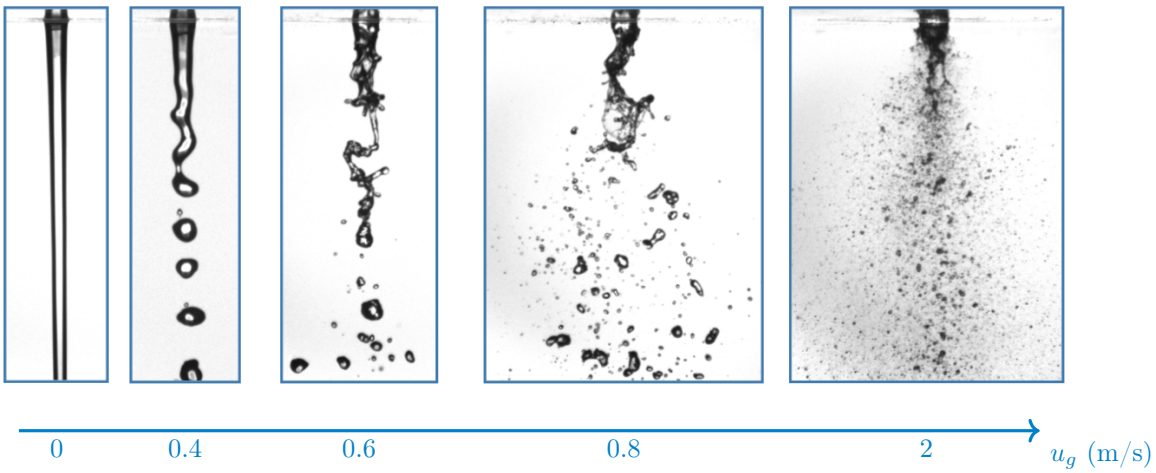


Figure 5: *Series of photographs of air assisted injection with co-axial jets at different velocities of air. Sources: CORIA*

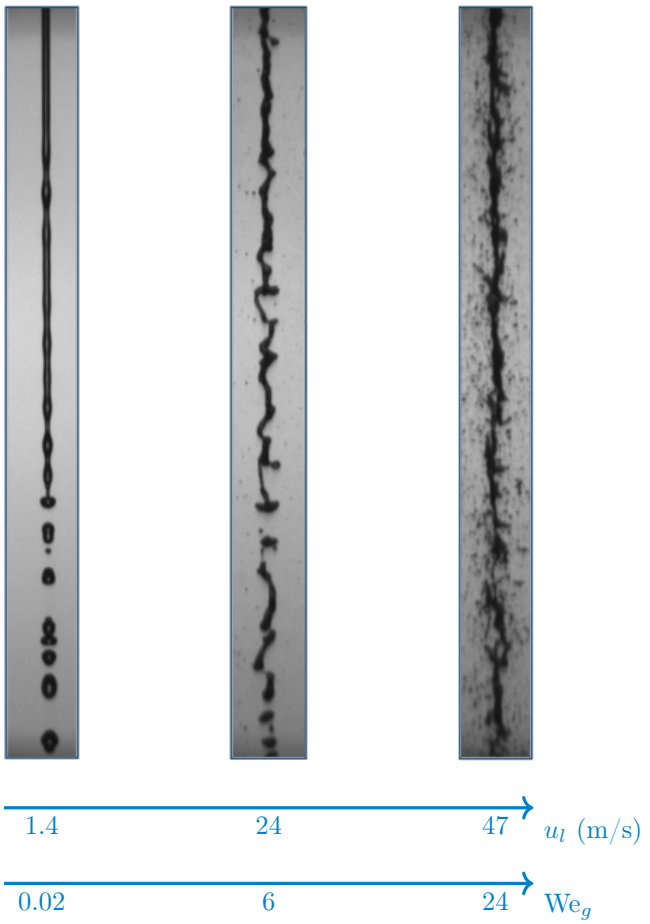


Figure 6: *Series of a free jet injections with different inlet liquid velocities. Sources: CORIA*

configurations. In such configurations, the liquid phase is introduced in the chamber at high pressure and high velocity:

$$p_l \approx 1.0 \cdot 10^8 \,\text{Pa} \quad \text{and} \quad u_l \approx 1.0 \cdot 10^2 \text{m}\,\text{s}^{-1}.$$

Typical Reynolds and Ohnesorge numbers are:

$$\text{Re} \approx 1.0 \cdot 10^4 \quad \text{and} \quad \text{Oh} \approx 5.0 \cdot 10^{-2}.$$

In these conditions, the instabilities lengths and the drops sizes, issued from the primary break-up, are at the order of 10^{-5} m. After the secondary break up the droplets can be of a size as small as a few micrometers (10^{-6} m). The nozzle diameter is around 10^{-3} m and the combustion chamber about 10^{-1} m (Lee and Park (2002)). Turbulent structures also interact with the jet and the droplets, and a typical size is given by the Kolmogorov scale. Although all these data are only rough estimations (the exact data strongly depend on the configurations), and only about spatial quantities, one can see that a full description of the injection process involves a wide range of physical scales.

These problems are called **multi-scale** problems and their resolution raises difficulties, since all scales have to be taken into account in a modeling approach for predictive simulations. For example, in Doisneau, Sibra, Dupays, Murrone, Laurent, and Massot (2014), it is shown that modeling the interactions between the cloud of droplets and acoustic phenomena necessitates an accurate description of the dispersion in size of the droplets.

Remark 1. *Recently, the standard view of liquid atomization has been questioned Dahms and Oefelein (2013), due to the pressure and temperature conditions, close to the transcritical regime in the new*

configurations of Diesel engines. Although the instabilities mechanisms are different in this regime, the injection can still be seen as a multi-scale two-phase flow with similar issues.

0.3 Challenges in the modeling of atomization

0.3.1 Different classes of two-phase flows

The question of modeling multi-scale two-phase flows can be approached in two ways:

- one can consider models which are valid at the finest scales of the problem. When describing the evolution of an interface, such models must predict the motion, deformation, break-ups and coalescence of this interface. In the case of the atomization of a liquid jet, the typical structures of the interface may have sizes from 10^{-3} m close to the nozzle exit down to 10^{-7} m for the smallest droplets. Such approach must then be valid at scales that are below 10^{-7} m,
- one can take advantage of a certain *homogeneity* of the finest structures and develop an appropriate reduced-order model. This model will be valid at a scale larger than the finest scales. In two-phase flows, these structures correspond to different possible topologies of the interface.

In Ishii (1975) and references therein, different categories (or **classes** or **regimes**) of two-phase flows are identified. Among these categories, one has the **separated-phase** flows, where both phases cover large and continuous spaces, and their separation is defined by a unique or by few portions of an interface. Another category is the **disperse-phase** flow where one of the phases is composed of an extremely large number of topological objects, surrounded by the other, continuous, phase. Taking the atomization process as an example, a separated-phase flow corresponds to the region close to the nozzle exit, while the cloud of droplets enters the category of disperse-phase flows (see Figure 3). The transition zone is called **mixed** or **transitional** flow in Ishii (1975).

Models valid at the finest scales of the problem will be used for DNS. For two-phase flows, this approach consists in solving continuum equations of fluid mechanics at any location, with fluid parameters, like density, viscosity or state laws, depending on the phase. The interface is considered as a sharp discontinuity, and jump relations across the interface are used to describe mass, momentum and energy exchanges. The essential point of DNS is that the interface is assumed to be exactly located. Now, when looking at the orders of magnitude of the previous paragraph, the DNS approach would require an accuracy up to 10^{-7} relatively to the typical scale of the whole domain. This approach is then interesting for the study of poorly understood interfacial phenomena (like the evaporation of the droplets, drag forces, etc), but such a problem size is not affordable for numerical simulations of the whole injection process in an industrial context. It may however be used for the description of the separated-phase zones and may be coupled with a more appropriate description of disperse-phase flows.

Reduced-order models for two-phase flows include different types of models. Some of these models are issued from the averaging of the continuum equations, in order to get a mean description at larger scales. These averaged equations may be used for both separated and disperse phase flows. Other models are based on a statistical description of the disperse-phase flows.

Let us give some more details about these different methods and discuss then some recent works that propose to combine approaches for both separated-phase and disperse-phase flows.

0.3.2 Interface tracking and interface capturing methods

We present here a series of methods which aim at modeling two phases that are separated by an interface, and in particular they aim at predicting the evolution of this interface. These methods are thus well-suited for separated-phase configurations and can be classified in two categories, according to their definition of the interface:

- the interface can be defined as a discontinuity, whose motion and evolution are resolved in the referential of its own motion (in a Lagrangian description): these methods are called **interface tracking** methods;

- the interface is a surface that is represented by a function. The values of this function are evolved through time in a fixed referential: these are **interface capturing** methods.

0.3.2.1 Interface tracking methods

The simulation of fluid dynamics often involves to discretize the space using a **grid**. The numerical solution is then given at the different points of the grid. In fully Lagrangian methods, this grid is moving and follows the fluid motion. It is then obvious that, without phase change (evaporation, condensation), the points that are initially located on the interface will stay on it, and may help locating this interface. One of the most known methods involving partially a Lagrangian description is the Front Tracking method (Tryggvason, Bunner, Esmaeeli, Juric, Al-Rawahi, Tauber, Han, Nas, and Jan (2001)), where the fluid equations are solved on a fixed grid, while a second moving grid is used to track the interface. The method leads to accurate results, but the combination of different grids may be complex to deal with in a numerical point of view.

0.3.2.2 Interface capturing methods

Among the interface capturing methods, the function used to implicitly locate the interface may be of different nature. It can be:

- a level set function that represents the distance separating a given point from the interface in the vicinity of this interface,
- a piecewise constant function whose discontinuity locus represents the location of the interface,
- a smooth function that experiences important variations in zones of the computational grid that will represent the interface location as transition zones (like in phase-field or Cahn-Hilliard models).

Let us note that with the second method, two options may be considered:

- the interface is assumed to remain a discontinuity and the method does only handle zones where the color function is 0 or 1. Often, like with VOF methods, the interface must be reconstructed,
- the color function is allowed to spread due to numerical diffusion, but must converge towards the real discontinuity when the diffusion is decreased. Also, the method must propose a coherent way of treating the diffusion zones.

Level-Set

In Level Set methods (see Osher (1988)), the implicit definition of the interface is given by a distance function (the distance from the interface) φ whose zero iso-values draw the interface (see Figure 7a). This function is advected by the fluid material velocity according to the following law:

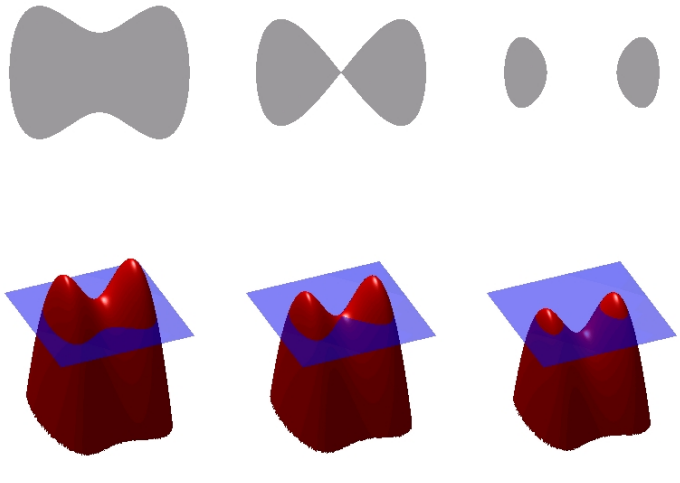
$$\frac{\partial \varphi}{\partial t} + \mathbf{u}^T \boldsymbol{\nabla} \varphi = 0. \tag{1}$$

Once this equation is solved over some time iterations, the level-set function must be rescaled, so as to stay a distance function (i.e. satisfying $|\boldsymbol{\nabla}\varphi| = 1$), at least in the neighborhood of points where $\varphi = 0$.

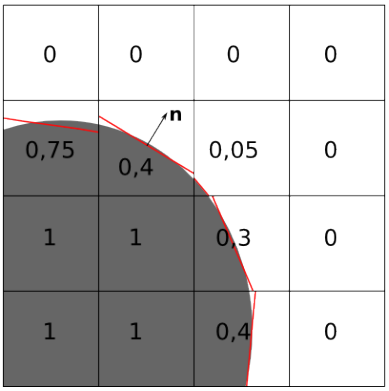
The advantages of the Level Set method are its ease of use and its accuracy for the interface description, provided that the scale of resolution is actually the scale of the finest interfacial structures. It is moreover easy to exploit the distance function so as to get the interface geometrical information (see Chapter 3). Among its shortcomings however, it is well-known that mass is not conserved when the level-set is rescaled and the method does not provide a model for the unresolved scales.

Volume-of-Fluid

The second widely used interface capturing method is the Volume-of-Fluid ((VOF) method (Hirt and Nichols (1981)). Now, the phase indicator is given by a color function C, satisfying the following transport



(a) Illustration of Level-Set method. *Source Wikipedia.*



(b) Illustration of VOF method. *Source Wikipedia.*

Figure 7: *Level-Set and VOF methods.*

equation:

$$\frac{\partial C}{\partial t} + \mathbf{u}^T \boldsymbol{\nabla} C = 0. \tag{2}$$

Then the interface is reconstructed in cells where $0 < C < 1$ so as to conserve the masses of both phases (see an illustration of the method in Figure 7b). In the case of incompressible fluids, C can be considered as the volume occupied by one of the phase, relatively to the volume of the grid cell. This quantity is then called **volume fraction**. Ensuring mass conservation comes down to conserve the volumes of both fluids through the reconstruction of C. While VOF method ensures mass conservation, its use is often much more complex than the Level-Set method (Le Chenadec (2012)).

Multiple hybrid methods have also been developed to combine the advantages of Level Set and VOF methods: for instance the CLSVOF method (Bourlioux (1995); Sussman and Puckett (2000); Ménard, Tanguy, and Berlemont (2007)), or the hybrid Lagrangian-Eulerian methods of Le Chenadec (2012).

All these methods are used in a DNS context, and necessitate high resolution so as to be reliable: it is indeed not ensured that these approaches are able to capture all deformations when the interface undergoes strong wrinkling. However, in many cases, or for high-resolution simulations, they provide valuable information on the interface topology, like the surface area density, which is an essential parameter in combustion problems.

Let us also mention that compressible methods with interface reconstruction have also been developed as in Jemison, Sussman, and Arienti (2014) for example.

Diffuse interface and phase-field models

Another approach considers that the interface has in fact a thickness of controlled width. Within this interfacial zone, a *phase-field* variable evolves continuously and according to an advection-diffusion law, so as to satisfy a Cahn-Hilliard type equation (Cahn and Hilliard (1958); Cahn and Hilliard (1959a); Cahn and Hilliard (1959b)). The interfacial zone must be kept at a constant width δ and in the limit $\delta \to 0$, the sharp interface solution must be recovered. Let us mention that diffuse interface methods using directly the strong variations of the medium density have also been studied using a van der Waals equation of state for describing both phases of the flow. One can find more details on these methods in Jacqmin (1999); Jamet, Lebaigue, Courtis, and Delhaye (2001); Jamet (2010).

Interface capturing methods based on two-fluid or mixture models

Contrary to the sharp or diffuse interface methods, the interface capturing methods do not need the interface to be infinitely thin or of controlled width, since they solve two-fluid or mixture models. In these models, the equations of both fluids are solved, assuming that at $(\mathbf{x}, t)$ one can find either a pure fluid, or a mixture of both fluids. The physical discontinuity between the phases is assumed to lie in a mixture zone, which is created by the numerical diffusion. However, the method must converge to the exact discontinuity when the numerical resolution is finer. This interface capturing method acts in the same way as many shock capturing methods, that create smooth but strong gradient zones converging towards the physical shocks as the space discretization tends to zero.

Dealing with this spread interfacial zones requires to provide constitutive laws for the medium in these zones. One often refers to **mixture laws**, since these laws may be derived from real mixture models (in the same way as the averaged systems of equations). Let us precise that the two fluids are still immiscible: the mixture zone is considered as a sub-scale set of continuous volumes of the two fluids.

For example, an interface capturing model that will be encountered throughout this thesis is the homogeneous and isothermal two-phase model:

$$\left\{\begin{array}{rcrcl} \partial_t \rho & + & \operatorname{div}(\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho Y) & + & \operatorname{div}(\rho Y \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \operatorname{div}\left(\rho \mathbf{u}\mathbf{u}^T + p\,\mathbf{Id}\right) & = & 0 \end{array}\right. \tag{3}$$

where both fluids share the same pressure p and same velocity $\mathbf{u}$. The mixture density and the mass fraction of one of the fluids will respectively be ρ and Y (the other mass fraction being $1-Y$). Initially, both phases are identified by the value of Y field, which is 0 in one of the phase, and 1 in the other. The numerical resolution leads to the apparition of a zone where $0 < Y < 1$: this is the mixture zone, and the real interface lies within this zone. This method is used in several works on separated-phase flows (Chanteperdrix, Villedieu, and Vila (2002); Grenier, Vila, and Villedieu (2013); Bernard-Champmartin and De Vuyst (2014)) and has the following advantages:

- the method does not necessitate any reconstruction, which could be computationally expensive,
- numerical diffusion depends on the numerical scheme used to discretize the equations. Gaining in accuracy *simply* necessitates an increase of the accuracy of the numerical methods as proposed in Després and Lagoutière (2001); Kokh and Lagoutière (2010) and used in Bernard-Champmartin and De Vuyst (2014),
- the mixture zone may have a physical meaning, as we will see in Chapter 2, and is then suitable for the development of sub-scale models.

Summary of the separated-phase flows modeling

All the methods presented in this section have been designed to solve the evolution of an interface, and their level of accuracy must comply with the finest scales of the interface deformation. In cases of a large number of pieces of interface (like in disperse phase flows) or for extremely small deformations of the interface, using such methods could be computationally expensive. It could be envisioned to couple them with disperse-phase models, that will be presented hereafter. But this coupling also necessitates a certain compatibility between the approaches. Among the methods presented here, the interface capturing method using mixture models seems to be related to the two-fluid models describing disperse-phase flows, and may possibly be a good candidate for a coupling strategy.

0.3.3 Models and methods for disperse-phase flows

Let us consider a two-phase flow made of a carrier phase and a disperse phase, *i.e.* a phase composed of a very large number of particles. Two standard approaches may be used to model this flow category:

- one can consider a *macroscopic* point of view, where the equations provide the evolution of the flow mean quantities and the interfacial mechanisms have been averaged,

- one can consider a *mesoscopic* scale, where the particles are considered as punctual and their characteristics are internal variables.

The first approach leads to the derivation of two-fluid systems of equations, usually found in the modeling of bubbly flows. From the mesoscopic description, two methods can be considered. First, each particle is followed individually, this is the Discrete Particle Simulation or point-particle DNS. This method may help understand the interactions between the carrier phase and the particles by taking into account many mechanisms, such as heat transfers (Zamansky, Coletti, Massot, and Mani (2016)). However, due to the large number of drops that must generally be simulated, this approach is not affordable for industrial simulations and are not always well adapted to highly parallel computations, because of potential strong inhomogeneities in the particles repartition.

The second method is inspired from the kinetic theory for gases: an equation governs the evolution of a probability density function, that describes the flow of particles. The carrier phase is solved using classic continuum equations of fluid mechanics, and source terms account for the interactions with the disperse phase. This method provides efficient models for the description of *polydisperse* sprays of droplets, which is an important feature in several industrial flows as well as for combustion simulations.

0.3.3.1 Averaged two-fluid equations

The averaged equations method gives rise to the family of two-fluid models. For example, these models are used in Cheng, Drew, and Lahey (1985); Antal, Lahey, and Flaherty (1991); Ishii (1975); Drew and Passman (1999) for the description of bubbly flows. Alternative techniques of derivation, based on a variational principle, enable to recover similar equations as the averaged equations. These models have been used for the description of pulsating bubbly flows in Gavrilyuk and Saurel (2002); Gavrilyuk (2012). Another approach is used in the domain of the combustion of granular media, where the form of the equations is postulated, and closure terms are derived so as to satisfy the second principle of thermodynamics (Baer and Nunziato (1986)). Chapter 1 will review in more details these different derivation techniques.

Usually, the averaged models rely on strong assumptions concerning the sub-scale structures of the disperse flows. For example, all bubbles are spherical and of the same size. In some works, an equation on the mean surface area density is added to the usual two-fluid equations and provides further information on the interface topology. In Drew and Passman (1999); Drew (1990), equations on the mean geometrical properties of the interface are also proposed. However, their closure still necessitates strong assumptions on the topology of the interface. Moreover, in the case of sprays of droplets, the range of the droplets sizes may be very large. This **dispersion** in size is better handled by the kinetic-based models.

0.3.3.2 Kinetic-based models

Within the category of kinetic-based approaches, one can find a large variety of methods. All these methods are based on the definition of a number density function (NDF), that describes the statistics of the population of particles according to the value of their internal variables. A population balance equation, the Williams-Boltzmann equation (WBE), rules the evolution of this population of particles. The NDF is defined over a space whose dimension is usually higher than the dimension of the physical space. Indeed, the evolution of the population of drops will depend on its distributions in size, velocity, temperature, etc. and on the evolution of these internal parameters. For example, evaporation effects reduce the sizes of the droplets and thus, the distribution in size is modified. The **phase space** is the name of the space of all possible values for these variables, called **phase variables**. Solving WBE in all its phase variables dimensions is often very expensive. Consequently, one can consider the following approach to solve, or approximately solve, WBE:

- when the space dimension is not too large (few phase variables), WBE can be solved directly. This is done for example in plasma physics with Vlasov equation (Grandgirard, Sarazin, Angelino, Bottino, Crouseilles, Darmet, Dif-Pradalier, Garbet, Ghendrih, Jolliet, Latu, Sonnendrücker, and Villard (2007)),
- the stochastic Lagrangian approach (Bird (1994)) is based on samples of representative particles,

which are tracked using a Lagrangian description. Although this method is accurate and probably the most widespread for disperse-phase flows, its shortcomings come from its need of a very large number of particles to converge statistically and the complexity to balance the work load between the computing units in parallel computational architectures,

- the sectional or multi-fluid method consists in discretizing the NDF along its size dimension (Laurent and Massot (2001); de Chaisemartin (2009); Laurent, Sibra, and Doisneau (2016); Nguyen, Laurent, Fox, and Massot (2016)). WBE is integrated over the whole phase space in each size section (which is bounded by a minimum and a maximum value in each section). Simplifying assumptions (one velocity per size section) allow to derive a system of equations for the number density of droplets, their mean momentum and mean energy, in each section. The sections may exchange mass, momentum and energy with each other and with the carrier phase,
- the moments method makes use of equations on high-order moments of the NDF. This method permits an accurate description of **polydisperse** flows (when the dispersion in size of the population of droplets is large), but requires closure efforts, either quadrature techniques (McGraw (1997); Fox, Laurent, and Massot (2008); Yuan, Laurent, and Fox (2012)) or with a reconstruction of the NDF satisfying an entropy maximization (Massot, Laurent, Kah, and de Chaisemartin (2010a); Kah (2010); Emre, Kah, Jay, Tran, Velghe, De Chaisemartin, Laurent, and Massot (2015)),
- the last two methods have been combined in hybrid multi-fluid methods: in each size section, a few high-order moments of the NDF are considered. This method shows an accurate description of the spray while reducing the number of size sections (Laurent, Sibra, and Doisneau (2016)).

Moreover, a new high-order moments method involving fractional moments has been introduced in Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016); Essadki, de Chaisemartin, Laurent, and Massot (2016). In the following paragraphs, we are presenting this method in more details, since it presents interesting properties in an eventual coupling strategy.

Number Density Function and Williams-Boltzmann Equation: we consider a set of N particles, droplets for instance. A particle k, $k = 1, ..., N$ is represented by a point, that can be its center of mass, with coordinates $\mathbf{x}_k$. The point evolves with a velocity $\mathbf{u}_k$, and is characterized by internal variables like the surface area of the particle S_k, its temperature T_k, etc. Let us remark that for spherical particles, S_k is characteristic of the size of the particle. For each particle, we define its internal phase space $\boldsymbol{\xi}_k = (\mathbf{x}_k, \mathbf{u}_k, S_k, T_k, ...)$.

Let us consider an infinitely large number of disperse-phase flow realizations for the N particles, in similar conditions. The mean realization is shown to converge to a normalized N-particles distribution function $f^{(N)} : (t, \boldsymbol{\xi}_1, ...\boldsymbol{\xi}_N) \mapsto f^{(N)}(t, \boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_N)$ such that:

$$\int_\Gamma f^{(N)}(t, \boldsymbol{\xi}_1, ...\boldsymbol{\xi}_N) d\boldsymbol{\xi}_1 ... d\boldsymbol{\xi}_N = 1. \tag{4}$$

This function satisfies the Liouville equation (Liboff (1969)). Now, we can assume that the particles do not interact with each other, but only with the carrier phase: this allows to reduce the definition of $f^{(N)}$ to the definition of a single-particle distribution function $(t, \boldsymbol{\xi}) \mapsto f^{(1)}(t, \boldsymbol{\xi})$ and to set:

$$f^{(N)}(t, \boldsymbol{\xi}_1, ..., \boldsymbol{\xi}_N) = \prod_{k \leqslant N} f^{(1)}(t, \boldsymbol{\xi}_k).$$

One finally defines the Number Density Function (NDF) f as:

$$f(t, \boldsymbol{\xi}) = N f^{(1)}(t, \boldsymbol{\xi}). \tag{5}$$

With the same assumptions of the independence of couplings between the particles, the NDF can equivalently been defined by:

$$f(t, \boldsymbol{\xi}) = \left\langle \sum_{k \leqslant N} \delta(\boldsymbol{\xi} - \boldsymbol{\xi}_k) \right\rangle, \tag{6}$$

where $< \cdot >$ denotes the average over all realizations, δ is the Dirac delta function. Then, $f(t, \boldsymbol{\xi}_k)\mathrm{d}\boldsymbol{\xi}_k$ is the probability of finding a particle with a state $\boldsymbol{\xi}$ in the piece of phase space $\mathrm{d}\boldsymbol{\xi}_k$ at time t.

From Liouville equation for the N-particles distribution, one can first derive an equation for the one-particle distribution function, and then Boltzmann equation for the number density function f:

$$\partial_t f + \partial_{\boldsymbol{\xi}} \left(\mathbf{F} f\right) = \mathcal{C}(f, f),$$

where $\mathbf{F}$ stands for the effects applied to the particles which modify its phase state, and $\mathcal{C}(f, f)$ is a collision operator, valid for binary collisions between the particles. This equation is used in the kinetic theory for gases. In the specific case of disperse-phase flows, one considers rather the Williams-Boltzmann equation (WBE) that writes Williams (1958):

$$\partial_t f + \partial_{\boldsymbol{\xi}} \left(\mathbf{F} f\right) = \mathcal{Q} + \Gamma,$$

where $\mathcal{Q}$ is a collision and coalescence operator and Γ is a break-up operator. When the phase variables of a particle are its location $\mathbf{x}$, velocity $\mathbf{u}$, size S and temperature T, then $\boldsymbol{\xi} = (\mathbf{x}, \mathbf{u}, S, T)$ and WBE has the form:

$$\partial_t f + \mathrm{div}_{\mathbf{x}}(\mathbf{u} f) + \mathrm{div}_{\mathbf{u}}(\mathbf{F}_{\mathbf{u}} f) + \partial_S(R_S f) + \partial_T(H_T f) = \mathcal{Q} + \Gamma, \tag{7}$$

with:

- $\mathbf{F}_{\mathbf{u}}$ the acceleration forces (for instance due to drag),
- R_S the contributions to the size changes (for instance due to evaporation),
- H_T the heat transfers.

The direct resolution of (7) in three spatial dimensions involves eight phase dimensions: three for the position, three for the velocity, one for S and one for T. To reduce the phase space dimension, simplifying assumptions can be made. For example:

- one can assume that the droplets are of the same size: this is the **monodisperse** assumption, in opposition with the **polydisperse** one,
- one can assume that the droplets of the same size move with a same velocity: this is the **monokinetic** assumption, in opposition with the **polykinetic** one.

In the following, we will be interested in the dispersion in the droplets sizes and present below a method, based on the moments of f, which can be used to solve (7).

Moments-based methods: first, let us define the **moments** of f. Let us assume that the phase space variables are $\boldsymbol{\xi} = (\mathbf{u}, S, T)$ ($\mathbf{x}$ is always resolved), and define $\mathcal{M}^{i_1,i_2,i_3,j,k}(t, \mathbf{x})$ the moment of order i_1 in the x-component of velocity (u_1), i_2 in the y-component of velocity (u_2), i_3 in the z-component of velocity (u_3), j in size (S) and k in temperature (T), by:

$$\mathcal{M}^{i_1,i_2,i_3,j,k}(t, \mathbf{x}) = \int_{\mathbf{u}} \int_S \int_T u_1^{i_1} u_2^{i_2} u_3^{i_3} S^j T^k f(t, \mathbf{x}, \mathbf{u}, S, T) d\mathbf{u} dS dT. \tag{8}$$

Usually, the indices i_1, i_2, i_3, j, k are integers and the first order moments in each direction of the phase space define the mean values of velocity, size and temperature. When assuming that the liquid phase is incompressible (of phase density ρ_l) and its internal energy is proportional to temperature, with the specific heat capacity c_P, one has:

$$\begin{aligned}
\mathcal{M}^{0,0,0,0,0}(t, \mathbf{x}) &= n(t, \mathbf{x}), \\
\int\int\int \rho_l V(S) f(t, \mathbf{x}, \mathbf{u}, S, T) &= m_l(t, \mathbf{x}), \\
\int\int\int \rho_l V(S) \mathbf{u} f(t, \mathbf{x}, \mathbf{u}, S, T) &= m_l(t, \mathbf{x}) \mathbf{v}(t, \mathbf{x}), \\
\int\int\int \rho_l V(S) c_P T f(t, \mathbf{x}, \mathbf{u}, S, T) &= m_l(t, \mathbf{x}) e(t, \mathbf{x}),
\end{aligned}$$

with $n(t,\mathbf{x})$ the number density of droplets, $V(S)$ the volume of one drop given as a function of its surface area S, $m_l(t,\mathbf{x})$ the mass density of liquid (different from the phase density ρ_l, and the following relation holds: $m_l = \alpha\rho_l$ where α is the volume fraction of liquid), $m_l(t,\mathbf{x})\mathbf{v}(t,\mathbf{x})$ the mean liquid momentum and $e(t,\mathbf{x})$ the mean internal energy.

The high-order moments method consists in deriving the equations for a certain number of moments of the NDF. Closure relations are then necessary, since the integration of WBE (7) brings about multiple moments of the NDF per equation. To illustrate this, let us derive the equation on the first-order moment in size, assuming a monokinetic flow: $f(t,\mathbf{x},\mathbf{u},S,T) = n(t,\mathbf{x},S,T)\delta(\mathbf{u}-\overline{\mathbf{u}}(t,\mathbf{x}))$. Let us also assume that the evaporation term follows a d^2-*law*: $R_S = -K$, with K a constant and that heat exchanges, coalescence and break-up terms are zero: $H_T = \mathcal{Q} = \Gamma = 0$. The acceleration force is a drag force given by a Stokes law: $\mathbf{F} = \frac{\mathbf{v}_g - \mathbf{u}}{\tau(S)}$ with $\mathbf{v}_g$ being the gas velocity, $\tau(S)$ being a characteristic response time of the droplet to the Stokes drag force. Finally, let us assume that $\tau(S) = DS$, with D a constant. WBE equation now reads:

$$\partial_t f + \mathrm{div}_{\mathbf{x}}(f\mathbf{u}) + \mathrm{div}_{\mathbf{u}}\left(\frac{\mathbf{v}_g - \mathbf{u}}{DS} f\right) - \partial_S(Kf) = 0. \tag{9}$$

The multiplication of WBE (9) with S^k, $k > 0$, and then the integration over all S, T and $\mathbf{u}$ values gives the following equation for the moments $m_k = \int_S \int_{\mathbf{u}} S^k f$:

$$\partial_t m_k + \mathrm{div}_{\mathbf{x}}(m_k\overline{\mathbf{u}}) = \psi_k^+ - \psi_k^- - Km_{k-1}, \quad k > 0,$$

and

$$\partial_t m_0 + \mathrm{div}_{\mathbf{x}}(m_0\overline{\mathbf{u}}) = \psi_0^+ - \psi_0^-,$$

where $\psi_k^+ = KS_{\max}^k n(t,\mathbf{x},S_{\max},T)$, $\psi_k^- = KS_{\min}^k n(t,\mathbf{x},S_{\min},T)$. For the maximum droplets size, we expect the distribution to vanish. The source term ψ_k^+ is then zero. On the other hand, the minimum droplet size is $S_{\min} = 0$, then, for $k > 0$, $\psi_k^- = 0$. The last unclosed source term is finally $\psi_0^- = Kn(S_{\min})$. Among the possible approaches to close the equations, one can mention quadrature-based methods (Yuan, Laurent, and Fox (2012); Nguyen, Laurent, Fox, and Massot (2016)), or entropy maximization methods (Massot, Laurent, Kah, and de Chaisemartin (2010a); Kah, Laurent, Massot, and Jay (2012); Vié, Laurent, and Massot (2013); Emre, Kah, Jay, Tran, Velghe, De Chaisemartin, Laurent, and Massot (2015)).

In practice, four moments in size are enough to accurately describe an evaporating polydisperse flow (Massot, Laurent, Kah, and de Chaisemartin (2010a)) and the numerical methods associated with this method are shown to be robust.

Fractional moments method: a different form of the high-order moments method has been developed in Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016), based on fractional moments in size of the NDF. These fractional moments can be related to the Gaussian G and mean H curvatures, with the surface area S and the volume of bubbles V. The Gaussian and mean curvatures are quantities usually used to describe the geometry of a local piece of surface. One intuitive definition of these quantities may be given by considering a infinitesimal small piece of surface $\mathrm{d}\Sigma$, around a point $\mathbf{X}$ lying on this surface. In $\mathbf{X}$, one can define the normal $\mathbf{n}$ to the surface. Now, let us consider the set of the intersections of all planes containing $\mathbf{n}$ and $\mathbf{X}$: these intersections are 1D curves with a given curvature k (see an illustration in Figure 8). The **principal curvatures** k_1 and k_2 of $\mathrm{d}\Sigma$ in $\mathbf{X}$ are defined as respectively the minimum and maximum values of k. Then, the mean curvature in $\mathbf{X}$ is defined by:

$$H = \frac{1}{2}(k_1 + k_2),$$

and the Gaussian curvature is:

$$G = k_1 k_2.$$

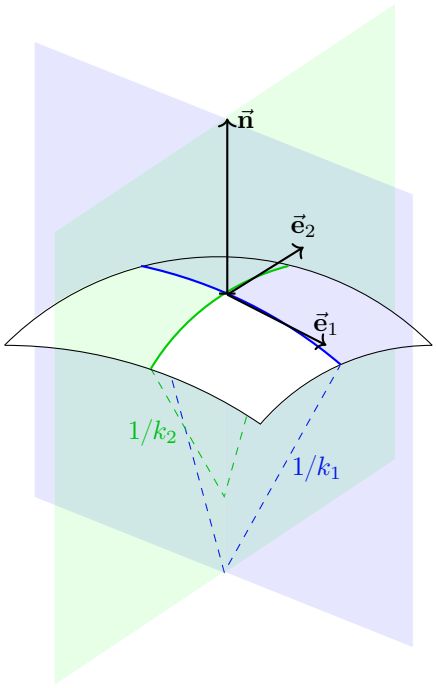


Figure 8: *Illustration of the local curvatures at* $\mathbf{X}^o$.

Now, by noting that, for a spherical droplet of surface area S, the mean and Gaussian curvatures respectively read: $H(S) = 2\pi^{1/2}S^{-1/2}$ and $G(S) = 4\pi S^{-1}$, and that the volume of a drop is given as a function of S by $V(S) = S^{3/2}/6\pi^{1/2}$, mean geometrical quantities of the flow (the mean *mean* curvature $\tilde{G}$, the mean Gaussian curvature $\tilde{H}$, the mean area density Σ and the volume fraction α) are related to four fractional moments in size of the NDF:

$$\begin{aligned}
\Sigma\tilde{G} &= 4\pi m_0,\\
\Sigma\tilde{H} &= 2\pi^{1/2} m_{1/2},\\
\Sigma &= m_1,\\
\alpha &= \frac{m_{3/2}}{6\pi^{-1/2}}.
\end{aligned}$$

From the integration of WBE (9) and using the monokinetic closure for velocity, the evolution equations of these moments are:

$$\left\{\begin{array}{lcccl}
\partial_t m_0 & + & \mathrm{div}_{\mathbf{x}}\left(m_0\overline{\mathbf{u}}\right) & = & -K\psi_0^-,\\
\partial_t m_{1/2} & + & \mathrm{div}_{\mathbf{x}}\left(m_{1/2}\overline{\mathbf{u}}\right) & = & -\dfrac{K}{2}m_{-1/2},\\
\partial_t m_1 & + & \mathrm{div}_{\mathbf{x}}\left(m_1\overline{\mathbf{u}}\right) & = & -Km_0,\\
\partial_t m_{3/2} & + & \mathrm{div}_{\mathbf{x}}\left(m_{3/2}\overline{\mathbf{u}}\right) & = & -\dfrac{3K}{2}m_{1/2},\\
\partial_t\left(m_1\overline{\mathbf{u}}\right) & + & \mathrm{div}_{\mathbf{x}}\cdot\left(m_1\overline{\mathbf{u}}\otimes\overline{\mathbf{u}}\right) & = & -Km_0\overline{\mathbf{u}} \quad + \quad m_0\dfrac{\mathbf{v}_g-\overline{\mathbf{u}}}{D},
\end{array}\right. \tag{10}$$

The terms ψ_0^- and $m_{-1/2}$, the last one being a negative moment, are closed by reconstructing the NDF $n(t,\mathbf{x},S,\mathbf{u},T)$ through an Entropy Maximization algorithm, originally proposed in Mead and Papanicolaou (1984) and adapted to the moments of system (10) in Essadki, de Chaisemartin, Laurent, and Massot (2016).

By interpreting some moments of the NDF as geometrical variables, this approach represents a first step in relating the geometrical description of interfaces with the description of polydisperse clouds of

droplets. In Chapter 3, we will study in more extent how a statistical description of interfaces can be used to describe any interfacial configuration.

0.3.3.3 Kinetic models for bubbles

Let us finish this review by mentioning works dedicated to the modeling of bubbly flows with the kinetic approach. Among the main differences between bubbles and drops is the compressibility of the first ones, while the second ones are usually considered as incompressible. Effects related to the bubbles compression, like pulsations, are generally not accounted for in standard kinetic-based models.

For a single bubble, Rayleigh-Plesset's equation (Rayleigh (1917); Plesset and Prosperetti (1977)) provides a good description of the evolution of the bubble volume. This equation indeed relates the time evolution of the bubble radius with the evolution of the pressure field in the surrounding liquid. However, this equation is not an evolution law for a flow of many bubbles. In the two-fluid system of equations derived in Gavrilyuk and Saurel (2002), an equation similar to Rayleigh-Plesset's equation models bubbles pulsations through an equation on the gas volume fraction field. In Chapter 2, we will see that this comes down to generalizing Rayleigh-Plesset's equation to a flow of monodisperse bubbles.

Kinetic-based models, as derived in Teshukov (2000); Teshukov and Gavrilyuk (2002), aim at generalizing Rayleigh-Plesset's equation for a set of bubbles, to get their collective behavior and their interactions with the liquid velocity field. The equation on the bubbles distribution function is close to Vlasov's equation used in plasma physics: the bubbles pulsations have a long-range effect on the liquid velocity field, which is considered as irrotational and divergence-free. The equations on the first-order moments of this distribution function are the hydrodynamic equations for the bubbly flow. However, in Teshukov and Gavrilyuk (2002), the simplifying assumption of monodisperse bubbles is made so as to close the equations.

0.3.3.4 Summary of disperse-phase flow modeling

The methods that have been presented in this section represent only a small sample of all the existing models for disperse-phase flows. However, they present interesting properties in a context of unified approaches, that may describe the whole injection process.

First, the averaged two-fluid models pertain to the same family of models as the models used in interface capturing methods. From the two-fluid model of Baer and Nunziato (1986), it is indeed possible to recover system (3) by assuming temperature, pressure and velocity equalities (see Kapila, Menikoff, Bdzil, Son, and Stewart (2001)). The transition zone of the interface capturing method using the 3-equation model could thus be considered as zones where the interface evolves and deforms at a sub-scale level. Averaged effects of this evolution and deformation could then be used to enrich the description of the mixture zones.

Second, kinetic-based models enable an accurate description of polydisperse flows. Moreover, moments-based models, and more particularly the fractional moments method, deal with quantities which are similar to the quantities found in two-fluid models (the mass density, the volume fraction, the area density). It could then be interesting to explore the physical meaning of the fractional moments in a more general flow configuration, and to couple them with variables used in separate-phase flow models.

0.3.4 Review of coupling approaches and atomization models

We have seen in the previous sections that the difficulties in modeling the whole flow of the atomization process is mainly due to the large range of physical scales that must be taken into account and to the difficulty to accurately localize the interface. On the one hand, interface tracking or interface capturing approaches are often used for separate-phase flow regimes, under the assumption that all scales are resolved. These approaches are hardly compatible with disperse phases, made of a large number of particles and a wide range in size of these particles. On the other hand, averaged models or statistical approaches are developed for the description of disperse-phase configurations. These two approaches

are very different, except when considering the mixture models used in interface capturing methods and which are close to the averaged two-fluid models.

To perform simulations of the whole atomization process, multiple options can be envisioned:

- a unified modeling approach, so as to get a reduced-order description of a two-phase flow independently of its class (separated or disperse),
- a coupling between existing models through source terms: the equations for both models are solved simultaneously and exchange information (mass, volume fraction, particles number...),
- a numerical separation of the zones where one or the other of the models must be solved (and then use boundary conditions at the interface between the two zones).

In the recent literature, some contributions aim at simulating two-phase flows characterized by the presence of separated-phase and disperse-phase flows. Let us start with a brief overview of existing strategies involving Lagrangian approaches:

- in the ELSA code (Beau (2006); Lebas (2007)), a seven-equation two-fluid model, enriched with an equation on the surface area density, is coupled with the stochastic Lagrangian method for the description of the disperse-phase. The zones of validity of both models are delimited by threshold values assigned to the volume fraction and area density,
- in Herrmann (2008); Herrmann (2010b); Herrmann (2010a), an approach combining the Level-Set capturing method for the separated phases and the stochastic Lagrangian method for the disperse phase is presented. The transition between both methods is performed through a zone of higher resolution near the interface. This involves refining parts of the mesh into blocks of finer cells: the implementation of this method relies on block-based AMR techniques (see the introduction of Part III). This local finer resolution improves the accuracy of the Level-Set method: fine structures can be resolved, and those structures, that satisfy certain geometrical conditions, are then transferred to the disperse phase. The algorithms associated with this method show good results in terms of parallel performance (Herrmann (2010b)),
- a similar approach was used in Arienti, Li, Soteriou, Eckett, Sussman, and Jensen (2013), with different algorithms and modeling choices. In particular, the interfacial zones are solved using the CLSVOF method. The identification of the structures, which are good candidates to feed the disperse phase, is performed within a single block of cells. Therefore, the resulting method does not involve communications between the blocks and is simpler.

We can note that these last two coupling strategies use Adaptive Mesh Refinement methods (AMR) in order to improve the accuracy of the numerical solution. It seems that the approaches of Herrmann (2010b) and Arienti, Li, Soteriou, Eckett, Sussman, and Jensen (2013) allow a correct description of the primary break-up. However, they are hardly compatible with a description of a polydisperse spray of droplets that occur after the secondary break-up, except for a reduced number of configurations.

Now let us consider fully Eulerian methods, which either involve one of the methods based on the kinetic approach for disperse-phase flows or appear compatible with them.

Interface capturing method using a model for interfacial area

A first work issued from Mandumpala Devassy (2013); Devassy, Habchi, and Daniel (2015) uses an interface capturing method with the seven-equation two-fluid model from Baer and Nunziato (1986); Saurel and Abgrall (1999) to solve the fluid equations for both phases. The objective of the modeling approach is to propose models for the primary and secondary break-ups in turbulent configurations: turbulence is modeled using Reynolds Averaged Navier Stokes (RANS) equations. The two-fluid system of equations is then completed by evolution equations on two liquid volume fractions α_s and α_d, and three interfacial area variables Σ_{mean}, Σ_s and Σ_d, inspired by Morel (2007). These five extra variables are associated with the different two-phase regimes: α_d and Σ_d are the volume fraction and surface area density for the disperse phase, α_s and Σ_s are the volume fraction and surface density area associated with the main liquid core structure. Moreover, to better model the interface deformation, a third surface variable is used. This quantity, Σ_{mean}, represents the surface area that the liquid main core structure would have without its ligament structures. The difference, $\Sigma_s - \Sigma_{mean}$, represents the surface area of

the ligaments, that are generated by the interface instabilities. It is used in a primary break-up model to determine the size of the drops that are detached from the main core and that feed the disperse phase. In Devassy, Habchi, and Daniel (2015), the primary break-up model is mainly based on a critical Weber number that also determines the size and the number of the detached drops, according to the experimental works of Marmottant and Villermaux (2004); Villermaux (2004). The equations on α_s, α_d, Σ_{mean}, Σ_s and Σ_d involve many terms accounting for:

- the transferred quantities from one phase regime to the others,
- the mean stretching effects for the surface density variables,
- the turbulent effects, that also contribute to surface production.

Contrary to most interface capturing methods, this work assumes that the finest physical scales cannot be solved numerically. Then it necessitates to model the sub-scale structures. This is achieved by the equations on the volume fraction and area density variables. In particular, the strong influence of the turbulent flows on the ligament structures is taken into account by the model. From experimental correlations, the sub-scale models provide a first estimation of the number and the size of the drops that are generated through the primary atomization. For secondary break-up, only information related to the area density is provided. Although the model does not hold a precise information on the dispersion in size of the disperse phase, one can note however that α_d and Σ_d variables are among the first-order fractional moments of a number density function, as defined in the previous sections. It seems that the modeling approach could be compatible with a finer description of the interface topology, for example by using the notions of curvatures.

Nevertheless, a significant modeling effort may be necessary to remove the contribution of the experimental correlations and obtain a parameter-free approach to predict the dispersion in size of the drops. Moreover, using multiple volume fraction variables can lead to situations where their sum is greater than one. These equations must then be handled and solved cautiously.

A coupling of an interface capturing and a sectional method

A second work (Le Touze (2015)) has proposed a coupling between two methods that are specifically devoted to the simulation of separated phase and disperse phase flows. A simple 3-equation mixture model, (similar to equations (3)) is used to solve the separated-phase flow evolution, while a sectional method (Laurent and Massot (2001)) is used to model the dispersion in size of the droplets. Source terms stand for break up mechanisms and for a *pseudo-coalescence*. They are used to model the transfers of mass, momentum and energy between the liquid phase of the mixture model and the sectional model. These source terms are active only in certain flow configurations: the break-up terms are activated in zones of low liquid volume fraction, while coalescence terms are activated in zones of low gas mass fraction. This prevents both effects to compensate. Indeed, in Le Touze (2015), the expressions for these effects are given empirically and according to experimental correlations and Le Touze (2015) proposes to consider that the mixture zones involve sub-scale interface deformation. The intensity of these deformations and characteristic instabilities frequencies are estimated using local Weber numbers and empirical correlations. In a proof-of-concept context, they also consider that the exchange source terms only affect the section of largest diameters in the disperse-flow model. Drops are indeed generated with a diameter that corresponds to the experimental setup, reproduced in the simulation: for example the cryogenic combustion research test bench Mascotte of ONERA (Le Touze (2015)). Let us finally mention that the numerical accuracy of the interface capturing method does not involve any interface reconstruction, and is increased by a new second-order scheme for unstructured grids (Le Touze, Murrone, and Guillard (2015); Le Touze (2015)).

The method proposed in Le Touze (2015) relies on simple modeling choices like: monodisperse drops production driven by the break-up processes, single velocity value for these created drops, etc. Nevertheless, it delivers a convenient framework that allows many improvement possibilities. Let us mention, for example, a better modeling of break-up and coalescence for enabling polydisperse models. This could be performed by using an evolution equation for the surface area density. Other improvements may be obtained by accounting for a bubbly phase into a richer two-fluid model (Cordesse (2019)). Finally, by coupling two Eulerian methods, it also shows robustness and simplicity. However, the computational

performances do not show to be very good.

However, this approach has several important flaws:

- the compressibility of both fluids, inherent of the mixture model, causes some problems when mass is added or removed from the liquid phase, generating pressure instabilities,
- while the sectional method for the disperse phase seems to be appropriate in this configuration, the approach does not allow a coupling with moments methods. Indeed, there is no relation between the separated-phase variables and any moments of a number density function. The moment methods can not be used, neither does the hybrid multifluid method,
- in relation with both previous points, one can note that the exchange terms are based on mass exchange between the two models. The conversion from mass to a geometrical description of the drops (to the size of the spherical drops) strongly relies on the incompressibility assumption. However, the liquid phase is assumed to be compressible in the separated-phase model. We can then wonder whether the mass exchanges from the separated-phase model to the disperse-phase model can be used in more general configurations,
- multiple volume fractions ($\geqslant 3$) are involved in this method, like for the method of Mandumpala Devassy (2013). Although much care is taken to *separate* the disperse phase from the pure liquid phase, one cannot completely discard situations involving a superposition of both phases, for instance when the exchange terms between the models are not very intense. This may lead to inconsistency in the system.

Other approaches

Let us finally mention other coupling strategies that have been explored recently.

In Fleau, Mimouni, Mérigoux, and Vincent (2015) (see also references therein), a three-field method is used to simulate bubbly flows. It consists in solving a system of equations similar to a two-fluid model, but with a supplementary field accounting for the disperse phase. The transfers of mass from one field to the other are modeled through exchange source terms. In particular, the transfer from the continuous gas phase to the disperse one occurs when the local curvature of the gas-liquid interface (computed through the gradient of the volume fraction field and that plays the same role as the color function in VOF approaches) is above a certain value. This value corresponds to a limit radius of the bubble under which its resolution as a separate-phase flow is no longer accurate. This threshold value is then related to the grid size. Some strong modeling hypotheses are made, such as imposing pressure equilibrium between all fluids and the incompressibility of the flow. Moreover, the disperse phase is assumed to be made of monodisperse bubbles. Let us mention also that no interface reconstruction is considered, but an interface sharpening method (see Tiwari, Freund, and Pantano (2013)) is used to prevent its diffusion.

With a very different approach, the method that is described in Doisneau, Arienti, and Oefelein (2017), studies the possibility of using kinetic-based models to describe the whole injection process, under an assumption of dense-particle flows near the nozzle exit. The gas phase is modeled using compressible Navier-Stokes equations, while the liquid spray is described by a multi-fluid model and two moments per size section. The coupling between both phases is achieved by exchange source terms. In Doisneau, Arienti, and Oefelein (2017), a comparison with a DNS simulation using the CLSVOF method is performed for the Spray A configuration (Bardi, Payri, Malbec, Bruneaux, Pichett, Manin, Bazyn, and Genzale (2012)). Although the polydispersion of the liquid phase was imposed at the beginning of the jet (according to a size distribution that was given by the DNS simulation), the new method shows promising results and the numerical scheme implementation presents good parallel performances.

Summary

In the works presented previously in this section, the use of mixture or two-fluid models appears regularly and seems convenient to describe interfacial zones. Indeed, these models enjoy several attractive features: they can involve compressible materials and they can fairly describe separated phases, provided that an accurate numerical scheme is used. They can also be enriched in order to describe interface instabilities and primary break-ups. This primary break-up phenomenon is indeed the first key phenomenon of the

atomization while the secondary break-up produces a strongly polydisperse disperse-phase. Polydispersion is well modeled by kinetic-based approaches, especially by equations on moments of a Number Density Function. Consequently, it seems important that the models used for the primary and secondary break-ups are able to *communicate* with moments-based models. This means that enough information from the break-up must be considered so as to describe the dispersion in size of the drops. This aspect is missing in most of the works that have been presented above, where models resort to external parameters.

A second aspect that emerges from some of the reviewed works is the effort of increasing the numerical resolution, by using dynamically adaptive grids (or AMR) or by using interface sharpening methods. Some numerical choices are naturally related to the modeling or coupling approaches. These choices also influence the computational costs of the methods and their applicability for computations on parallel architectures.

0.4 Discretization strategies in a HPC context

In the previous section, we have discussed the different models, or coupling of models, that exist to describe two-phase flows. The second important aspect of numerical simulations is the way the equations of these models are discretized and solved numerically. Indeed, the accuracy of the numerical solution depends on the choice of the numerical scheme in two ways. First, it directly depends on the order of accuracy of the discretization (size of the mesh, truncation error of the scheme, etc.). Second, since a method which is well-suited for parallel computation can be used on supercomputers and handle larger problems sizes: more accurate solutions can be obtained.

0.4.1 Recall of some parallel computing aspects

Nowadays, the architectures of supercomputers are mainly described by a hierarchical structure, made of:

- a large number of computation **nodes**, linked by a fast communication network,
- the nodes are composed of a set of **processors** (often two processors per node), made of several CPU **cores** (traditionally around 12 cores per processor). The memory on a node is accessible to all its cores,
- recent processors may be composed of much simpler cores (they are called then **manycore** processors) or may contain accelerators such as Graphic Processing Units (GPU). These processing units are designed for arithmetic intense computations, that do not require much access to cache memory. Such architectures are called **heterogeneous** architectures.

Associated with the different structure levels presented above, different parallel computating paradigms can be used.

First, the work load of a computation can be distributed over **processes** lying on different nodes. Since the nodes do not share any memory, the processes must exchange data, so that the computation runs correctly. These messages are usually handled by a Message Passing Interface (MPI) paradigm (Gropp, Lusk, Doss, and Skjellum (1996)). MPI processes may also be defined within each node when each core is assigned an MPI process.

On each node, where the memory is easily accessible to all processing units, one can also use a *shared-memory* computing paradigm, such as OpenMP (Dagum and Menon (1998)) for example. There, the work load is again distributed between the processes, under the form of **threads**. There is no need for communications between the cores, since the results of their computation updates the common memory space. However, with this method, one must take care of synchronizing the threads. Recent improvements of this paradigm consist in defining **tasks**, and then distributing these tasks between the computing units (Ayguade, Copty, Duran, Hoeflinger, Lin, Massaioli, Teruel, Unnikrishnan, and Zhang (2009)). This helps parallelizing pieces of codes, whose computation costs may be very different.

Computations on accelerators are prescribed when highly parallelisable and arithmetic intense tasks

must be achieved. Indeed, computing units on accelerators do not own much memory, but they are more numerous and more efficient than CPU cores on processors. Languages, such as CUDA can be used to perform computations on GPUs.

Finally, recent advances have been made to facilitate computations on heterogeneous architectures, by using both CPUs and GPUs capacities: for example, StarPU (Augonnet, Thibault, Namyst, and Wacrenier (2009)) enables to schedule the different tasks of a code and to send them to the appropriate computing units.

In the following, we will consider the classic MPI paradigm, that can be used on all supercomputers. Although wall-clock time may be gained with a parallel computation compared to a sequential one, the following points must be considered to improve the efficiency of the application:

1. the reduction of the total CPU time, *i.e.* the sum of the times spent on each processing unit,
2. the reduction of the memory consumption of an application,
3. a good scalability, *i.e.* the wall-clock time should ideally be divided by two when using twice more processes.

The first point may be achieved by optimizing the code. The second point is often more complex to deal with, since the prediction of the memory consumption is still an open issue (Rozar (2015)). Finally, among the factors that may go against the third point, one has:

- many communications between the processes. Indeed, the communication times are not decreasing when the number of processes increases, on the contrary, according to the type of communication, they can increase,
- work load imbalance between the processes, *i.e.* when some processes are assigned shorter tasks than the others, then they may have to wait for the other processes,
- memory accesses may also be expensive, when the cores must access zones of the memory that are not contiguous.

In our discretization strategy, we choose to take into account all these aspects. In particular, the space discretization strategy impacts the possibility of parallelizing the computation.

0.4.2 Discretization methods: from Lagrangian to Eulerian approaches

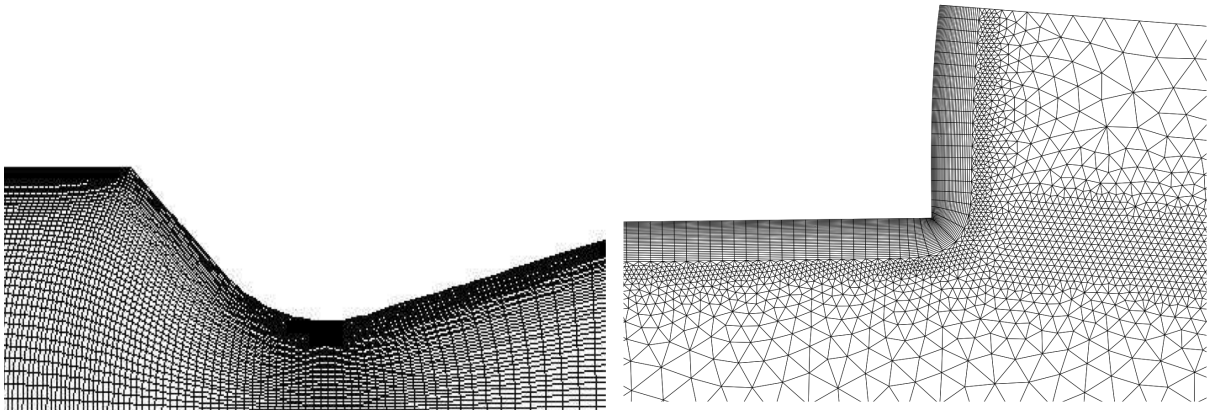
The time and space (and other phase space variables) discretization of systems of Partial Differential Equations (PDEs) can be made in very different ways. We present below different spatial discretizations.

Lagrangian grids

In the usual Lagrangian methods, discretized elements move and deform following the medium velocity field (von Neumann and Richtmeyer (1950)). With these approaches, the discontinuities of the numerical solution suffer little diffusion. However, when the flow is strongly sheared or has strong vorticity, the cells of Lagrangian grids are strongly deformed in long times and may lead to poor resolution. Moreover, for two-phase flows, where each fluid has its own velocity, such Lagrangian methods necessitate two grids, so as to track each phase. This two-grid system can be computationally expensive and complex to handle.

Mesh-free approaches

Another form of Lagrangian discretization is represented by the mesh-free methods, such as the *Smoothed Particle Hydrodynamics* (SPH) methods (Lucy (1977); Gingold and Monaghan (1977)). In SPH approaches, the fluid is represented by a set of particles, that interact with each other. These interactions are designed in order to recover the continuous equations of the fluid. In particular, properties are associated to each particle, such as mass, momentum, energy and a *smoothing length* over which the properties are *smoothened*. The particles evolution is then solved in a Lagrangian way. One of the advantage of SPH method is that its resolution level can be dynamically adapted, by adapting the smoothing lengths of the particles according to the flow configuration. For example, in zones of dense numbers of particles,



(a) Structured grid.

(b) Hybrid grid: structured near boundaries, unstructured otherwise.

Figure 9: *Examples of fixed structured and unstructured grids. Source Sibra (2015)*

this length can be shortened. Another way of adapting the resolution is by splitting the particles into new particles with shorter smoothing lengths. However, traditional SPH methods show large numerical viscosities and have difficulties to preserve sharp discontinuities of the flow, like shocks (see the discussion in Hopkins (2015) and the references therein).

Stationary grids

A stationary grid means that the grid does not follow the fluid motion, but is kept fixed during at least one time iteration. **Fixed** grids are grids that will not change during the whole simulation. On the contrary, **adaptive** grids can be changed from one iteration to the other by adding or removing grid elements.

For fixed grids, one usually identifies **structured** grids for which the connectivity of the elements is regular. For example in 2D, a grid made of only rectangular elements is structured (see Figure 9a). The elements of structured grids can be ordered, which simplify their use in practice. **Unstructured** grids however are made of elements whose connectivity is not regular, and that cannot be ordered (see Figure 9b). But unstructured grids enable to easily discretize domains with complex geometries, or to define zones of higher or lower resolutions. The solutions computed on fixed grids are generally less accurate than Lagrangian ones with respect to the fluid motion: sharp discontinuities may be artificially diffused. To counteract this issue, adaptive methods have been developed. Unstructured adaptive methods can be found in the literature (Holmes and Connell (1989)), but the structured methods are more widespread, through the Adaptive Mesh Refinement and Adaptive Multi-Resolution techniques. We present this last method in the next section. While parallel computations with fixed grids are usually performed by a domain decomposition that is defined initially and kept constant during the computation, the efficient parallelization of adaptive grids is not so obvious.

Hybrid methods

Let us also mention here methods that combine Lagrangian and Eulerian approaches, to benefit from some of their advantages. Among these hybrid methods, one can find the *moving grids* (Huang, Ren, and Russell (1994); Koltakov and Fringer (2013)), *Arbitrary Lagrangian Eulerian* (Trulio and Trigger (1961); Hirt, Amsden, and Cook (1974); Vazquez-Gonzalez (2016)) and a mixing of SPH and AMR methods (Hopkins (2015)). In moving grids methods, the grid is deformed so as to improve the spatial resolution in zones of interest. So the grid does not necessarily follow the motion of the fluid. In ALE methods, the mesh can be deformed, but under some regularity constraints. In indirect ALE methods, the grid follows the motion of the fluid in a Lagrangian way and is regularly reset to a regular grid. During this process, the numerical solution is then projected onto the new grid.

0.4.3 Structured adaptive grids methods

The idea of structured adaptive grid methods is to benefit from the adavantages of structured fixed grids methods (regular connectivity, application of finite volume or finite element schemes), enriched by the possibility of increasing or decreasing the resolution rate in certain zones of the domain, according to the needs for accuracy of the solution in these zones. Unlike fixed unstructured grids, the adaptive structured grids are changing throughout the computation.

Compared to fixed grids, the aim of these adaptive techniques is either to allow a finer simulation for approximately the same computational cost, or to reduce the computational cost while preserving the quality of the solution. We adopt this second point of view in the following, and will talk about **compressed** grids. The **equivalent uniform grid** is then simply defined, with respect to a compressed grid of smallest cells with size Δx, as a uniform structured grid of cell sizes Δx. The **compression error** is the numerical error introduced by the resolution of the equations on the compressed grid instead of using its equivalent uniform grid.

Now, there exist different approaches to determine which zones of the domain should be adapted:

1. a mathematical approach, called *adaptive multiresolution*, based on multilevel analysis of the regularity of the solution and which enables to control the compression error,
2. heuristic criteria based on the local stiffness of the solution,
3. criteria based on the truncation error of the numerical scheme, and its estimation at a coarser level.

The second and third approaches correspond to the Adaptive Mesh Refinement (AMR) approaches, and are discussed in Part III. Let us just mention here that the strong development of AMR codes and libraries has lead to highly scalable methods, well-suited for HPC simulations. For example the `p4est` library (Burstedde, Wilcox, and Ghattas (2011)), presented in Chapter 7, was shown to scale up to 458,000 cores.

Multiresolution methods have been developed with the will of controlling the numerical errors in a rigorous mathematical way. It is based on the decomposition of the solution on an appropriate function base, the **wavelets**, associated with a multilevel analysis: each base function carries information at a given level and a given location. For instance, if a solution does not present locally strong variations, then the decomposition over the function base will give significant coefficients for the first coarser levels and negligible values for the other levels. Wavelet decomposition provides information on the regularity of the solution in both spatial and frequency domains.

First uses of wavelets were made in the image and signal processing communities, where they are largely used to compress data and pictures. They have also developed in the CFD community, where they are especially useful for turbulent computations (see a review of these techniques in Schneider and Vasilyev (2010)).

A simple base of functions is made of piecewise constant functions, that are piecewise constant over a hierarchy of nested grids: this is the Haar wavelets base. In 1D, these grids are dyadic, in 2D they correspond to a quadtree-based grid, and in 3D, to an octree-based grid. The multilevel analysis starts from the finest of these grids, where the solution is given at its finest level. The error generated by the projection of the solution over the bases associated with one-level coarser grid is computed: the coarser solution is obtained from the initial one by an averaging process. These successive errors estimations, from one grid to a coarser one, are evaluated and stored. They are called *details* of the solution when considering its coarsest representation. Other wavelets bases can be used to increase the precision of the method (like the compactly supported orthonormal wavelets of Daubechies (1988)).

Among the methods that use the wavelet decomposition and the multilevel analysis is the adaptive multiresolution, based on the pioneering works of Harten (1994); Harten (1995), and developed by Cohen, Kaber, Müller, and Postel (2003); Müller (2003). There, the details issued from the multilevel analysis are compared with a threshold value: details that are below this threshold are not taken into account anymore. This allows to associate a level of resolution to the solution at each spatial location, while ensuring a controlled error. Since the level of the details correspond in fact to a spatial cell size, this is how the spatial adapted grids are generated.

In practical resolution of PDEs, and when associated with a numerical scheme, only one level of details is stored in each cell. On the one hand, at each mesh change, these details are re-evaluated, and cells where the details are now below the threshold can be coarsened. On the other hand, a prediction step is used to determine cells where the solution may vary a lot during the scheme updating process. These cells are then refined and the value set in the smaller cells is computed using an interpolation procedure. For this prediction step, the method requires a stencil of cells that is composed of multiple layers of neighbors. Considering the tree structure of the grid, this operation may be costly, due to the tree traversal, and may prevent to use this method in a distributed memory framework. However, adaptive multiresolution techniques show good results with shared-memory programming models (Descombes, Duarte, Dumont, Guillet, Louvet, and Massot (2017)). Applications of adaptive multiresolution can be found in Rossinelli, Hejazialhosseini, Spampinato, and Koumoutsakos (2011); Deiterding, Domingues, Gomes, Roussel, and Schneider (2009); Duarte (2011); Duarte, Massot, Descombes, Tenaud, Dumont, Louvet, and Laurent (2012); Descombes, Duarte, Dumont, Guillet, Louvet, and Massot (2017).

0.4.4 Summary

Each of the discretization strategies presented in this section has its own source of numerical errors (numerical diffusion for instance), or numerical difficulties (deformed cells for Lagrangian approaches for example). Different approaches have been developed to counteract these shortcomings, like hybrid Lagrangian-Eulerian methods, adaptive grids, etc.

The numerical schemes associated with structured grids are widely used and well-developed for the computation of fluid dynamics. Hence, it seems interesting to consider the structured adaptive grids framework, which enables to reduce the computational cost by compressing the grid on which the equations are solved. Among the strategies used to dynamically adapt the grids, although the multiresolution approach allows a rigorous control of the compression errors, it is not yet suited for distributed parallel computing, while recent developments of adaptive mesh refinement methods now show performance.

0.5 Overview of a new strategy for the modeling and the simulation of two-phase flows

0.5.1 Main idea of the new strategy

The works presented in this thesis are part of a desire to treat the issues of modeling and simulating the whole injection process, from the interfacial flows at the exit of the nozzle up to the polydisperse spray of the liquid droplets. We make the choice of developing a **unified modeling approach**. More precisely, we are looking for a reduced-order model, or a hierarchy of reduced-order models, that are handling the same physical quantities, whatever the flow regime. One classically considers a first set of flow parameters: the mass, the momentum and the energy of each component. A second set of parameters will help us to better characterize the flow regime:

- the volume fraction of one of the phase is a good indicator of a dilute disperse regime for instance, when one of the phases has a negligible volume fraction comparing to the other phase,
- the interfacial area density, that increases in zones of strong interfacial instabilities or in dense disperse phase regimes,
- geometrical information about the interface, that could be the radius of spherical particles (drops or bubbles) or in a more general context, the local curvatures of the interface. These geometrical information should also provide a description of the dispersion in size of the particles,
- the interfacial velocity, when the local evolution of the interface does not follow the dynamics of the local bulk phases.

Moreover, an appropriate statistics should be able to characterize the distribution of the topological interface configurations according to all these parameters.

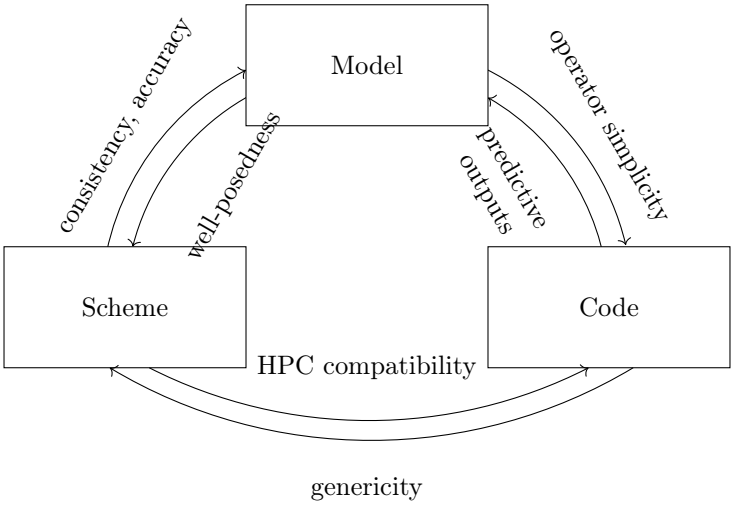


Figure 10: *Relations between the modeling, numerical and computational aspects. The constraint a part A owes to a part B is represented by an arrow from A to B.*

Each of these quantities requires an evolution equation. This equation may contain parameters, whose values are associated with the structure of the two-phase interface (for example the drag force coefficient, that depends on the size of the particle) and thus associated with the two-phase regime. The values of these parameters may evolve continuously from one regime to the other. They may also vanish (or become very large), which changes the structure of the systems of equations towards reduced equations. This hierarchical relation from one model to the other will be called *asymptotical consistency*.

Then, the associated numerical methods will have to satisfy the following criteria:

- they must be consistent and as accurate as possible with respect to the models (this also necessitates that the equations defining the models satisfy some mathematical properties, as we will see later),
- they must be consistent in the transitions from one flow regime to the other (they should have asymptotic preserving properties),
- they must be robust in the asymptotic limits or with respect to strong variations of the flow parameters (with strong gradients of density at the two-phase interfaces for example).

Moreover, we are looking for methods that are compatible with highly parallel computations. For example, these methods should be easily and equally distributed over a large number of processing units and should not rely on lots of communications.

Finally, the integration of these numerical methods in a unique code is considered. This code must then be designed to facilitate the integration of these new numerical methods, and possibly to allow a coupling of these methods.

In conclusion, we can say that each of the previously mentioned specifications must be fulfilled. However, it is important to keep in mind that these matters are connected and should be considered and investigated as connected issues. Examples of connexions between the three axes, that are the modeling, the discretization choices and the code development, are proposed in Figure 10. Therefore, in this thesis, we propose an approach combining all these aspects, where the modelling and numerical developments are regularly assessed with respect to experimental data.

0.5.2 Modeling aspects addressed in the present thesis

The modeling developments presented in this thesis are part of the general strategy, discussed in the previous paragraph. In particular, they are concerned with the study of the compressible two-fluid and mixture models, used in the interface capturing strategy without any interface reconstruction. We are

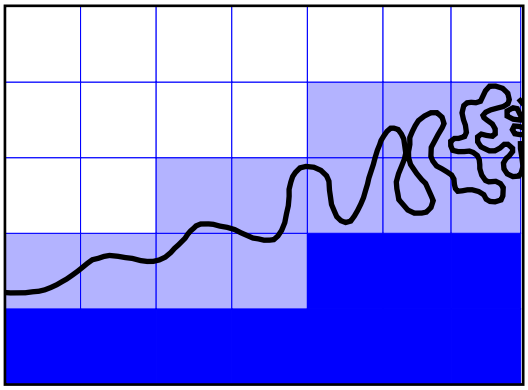


Figure 11: *Illustration of the numerical diffusion of a two-fluid (or mixture) model, and sub-scale representation of the two-phase interface.*

indeed looking for a modeling of the sub-scale effects that may develop in the zones where the interface is diffused (see an illustration in Figure 11).

By using a variational principle, we can recover the equations of such models and also of models that are enriched with more sub-scale effects. We present this technique of derivation, inspired by Gavrilyuk and Saurel (2002), in Chapter 1, and use it to derive a hierarchy of two-phase mixture models, that are asymptotically consistent. We moreover show that these models present interesting mathematical properties, that are helpful in the design of appropriate numerical schemes. We show that these models have a convective part that is hyperbolic and compatible with an entropy budget equation. This entropy evolution equation allows us to add dissipative terms that are compatible with the second principle of thermodynamics.

The derivation considered in Chapter 1 involves the use of some parameters to close the equations and ensure the entropy inequality. In Chapter 2, we propose physical identifications for these parameters, in the context of bubbly flows, where similar models are used and experimental data are available. From these identifications, we introduce a first information on the flow topology, that is the radius of the bubbles. This identification highlights also the physical meaning and the effect of pressure relaxation towards equilibrium, which is usually assumed to happen infinitely fast.

In Chapter 3, we shall study means of introducing more topological information in the models with variables that are associated with the characterization of the interface and its deformation. We shall highlight connections with kinetic-based models that will allow us to derive new terms to our Eulerian mixture models.

0.5.3 Numerical aspects of our strategy

Two-phase flows simulations and interface capturing methods often raise multiple numerical difficulties, as for example:

1. the high density ratio between the two components (for instance $\sim 1/1000$ for air-water systems),
2. the difference between the acoustic impedances, which are given by the product of density by sound velocity ($\sim 4.0 \cdot 10^2$ Pa $\cdot$ s/m for air, $\sim 1.5 \cdot 10^6$ Pa $\cdot$ s/m for water),
3. stiff equations of states (for mixtures near one-phase states),
4. stiff relaxation first-order source terms,
5. non-conservative equations,
6. the numerical diffusion of the interface,
7. the fluids compressibilities.

This list is not exhaustive: we could also mention the ratio of the viscosities or, for spray models, the presence of accumulation zones. With our discretization strategy, we propose to address some of the issues listed above, and to design numerical schemes that are robust, accurate and of affordable computational complexity.

The systems of equations derived in Chapter 1 are characterized by a conservative and hyperbolic convective part, that is supplemented by source terms that can be considered as relaxation terms. This structure of the equations enables the use operator splitting with a classic finite volume scheme for the convective part and standard solvers for the integration of the source terms. These methods are presented in the introduction of Part II and in Chapter 4.

However, some difficulties related to compressible two-phase flows involve an adaptation of the classic methods. First, as we will see in Chapter 4, the ratio of density and of sound velocity of a gas-liquid mixture may involve an overestimation of the CFL constraint. A modified scheme is proposed in this same chapter to tackle this problem.

Second, the source terms are strongly non-linear and very stiff, that makes their integration complex with standard methods. We propose in Chapter 5 a strategy to approximate this integration and ensure stable results. A special study of the source terms effect on the numerical solution and its asymptotic limit is also performed in Chapter 5, where we show limitations of the usual Riemann solvers presented in Chapter 4.

Third, we shall also deal with another specific problem involved with classic Godunov-type methods. It occurs when those methods are applied to compressible flows and concerns the simulation of low Mach number flows. This very critical issue has been widely studied (see Dellacherie (2010); Dellacherie, Omnes, and Rieper (2010) among others) and improvements of the Finite Volume solver introduced in Chapter 4 will be presented in Chapter 6. The resulting modified schemes will be tested against simulations of a dam break in Chapter 10.

Finally, Chapter 6 describes a series of numerical tools that allow to increase the accuracy of the numerical solution. This task will be achieved by using AMR methods, and more particularly *cell-based* AMR. This method involves the use of non-conforming meshes, for which classic high-order methods must be adapted.

0.5.4 Computational challenges and code development

The AMR techniques aim at saving computational resources by reducing the total work load and the memory consumption. However their use on High Performance Computing architectures is not an easy task, as we will see in the introduction of Part III. In this context, the p4est library proposes several solutions to issues related to highly parallel computations and AMR (see its detailed presentation in Chapter 7):

- p4est algorithms are designed to be highly scalable,
- the concept of *forest of trees* allows the p4est library to discretize domains with complex geometries,
- the p4est library is based on the *Inversion of control* principle, that makes it compatible with a wide number of applications.

We have developed a finite volume code, based on p4est. This development was achieved with the idea of a generic framework in which all future models and schemes could be integrated. The CanoP code and its first performance results are presented in Chapter 8.

Finally, Chapter 9 and Chapter 10 show numerical results stemming from classic one-dimensional Riemann problems to multi-dimensional simulations. They aim at validating the models and numerical methods presented previously as well as their implementation in the CanoP code. The comparisons of dam breaks of Chapter 10 will especially show that low-Mach configurations can be simulated with a compressible two-phase model on AMR grids.

Part I

Modeling: derivation of two-phase models for separated and disperse-phase flows

Abstract

In this part, we are interested in the derivation of two-phase models, which can be used either for the description of separated-phase or of disperse-phase flows. More specifically, we study two-fluid and mixture models. In Chapter 1, after an overview of some techniques of derivation for these models, we propose to use Hamilton's variational principle and the second principle of thermodynamics to derive a hierarchy of homogeneous two-phase models with sub-scale pulsation effects. These models are assessed in a context of bubbly flows in Chapter 2. Sub-scale parameters are identified and the regimes of validity for each two-phase model are studied. In Chapter 3, we propose to go further in the modeling of the sub-scales mechanisms, by studying first the geometrical parameters of interfaces and their evolution, and then by considering a statistical description of general forms of interfaces. Chapter 2 and Chapter 3 may also be found in Drui, Larat, Kokh, and Massot (2016). Chapter 3 is developed in Essadki, Drui, de Chaisemartin, Larat, Ménard, and Massot (2017).

Chapter 1

Derivation of a hierarchy of two-phase mixture models

We can define mixture models, as models that are based on the assumption that, given a two-phase flow, at any time-space point $(t, \mathbf{x})$, one can find either one of the two phases, either a *mixture* of them. For two immiscible fluids, a mixture zone can be considered in multiple ways:

1. it can be seen as a piece of volume around $\mathbf{x}$, composed by multiple sub-volumes made of only one of the two phases. The interface or pieces of interface that separate these volumes are not exactly located,
2. it can be a space location $\mathbf{x}$ where, during time $\mathrm{d}t$, both phases have gone through,
3. finally, it can be a space location $\mathbf{x}$ where, at a given moment t, the probability of finding either one phase or the other is not zero nor one. This probability is measured over multiple realizations of the flow in the same initial and boundary conditions.

The three visions presented above correspond in fact to three averaging operations, that are respectively space averaging, time averaging and **ensemble** averaging. In general, we will consider as mixture zones, control volumes of the flow that contain one or multiple pieces of interface.

This is why mixture models are employed on a one hand in interface capturing methods, for the separated two-phase flows regime. In this context, the mixture zone appears due to the numerical approximation of the equations. The interface that separates the two fluids lies in this mixture zone, but is not precisely localized, unless the numerical resolution is extremely accurate. This approach generally assumes that the whole physics of the flow is resolved, and some strong assumptions are made, such as pressure equality between the two fluids (= mechanical equilibrium), velocity equality (= kinematic equilibrium) or temperature equality (= thermal equilibrium) as in Chanteperdrix, Villedieu, and Vila (2002); Grenier, Vila, and Villedieu (2013); Bernard-Champmartin and De Vuyst (2014). The number of equations of these models is then quite low.

On the other hand, one can also use two-fluid models when the interface does not need to be resolved precisely, but collective physical effects due to a large number of fluid particles (thus, in the presence of a large number of pieces of interface) have to be modeled. The mixture has then the form of a disperse-phase flow for example, like a bubbly flow (Cheng, Drew, and Lahey (1985); Gavrilyuk and Saurel (2002)). But one could also think of a unique interface, that is strongly deformed, due to instabilities, and whose deformations can be of very small size in comparison with the typical size of the two-fluid flow. For these two configurations, the collective effects stand for physical effects which may occur across the interface, like surface tension effects, thermal exchanges or viscous dissipation, and which may depend on its form and dynamics, *i.e.* on its local curvature or its local velocity. However, as these effects occur at a scale, called the *sub-scale* or *microscopic* scale, which can not be resolved, they are accounted for in the equations after being averaged. The number of equations for these two-fluid models is generally six or seven. Moreover, the equations contain many averaged terms that make the systems complex to close and to solve, with sometimes ill-posedness issues.

The idea of this chapter is to study methods that can be used to enrich the models that are usually used for separated-phase flows, with physical sub-scale effects. These sub-scale effects should model the interfacial deformations and the interactions of both fluids in the mixture zones at the scales which are not resolved. Three approaches may then be considered:

- the first one is based on continuum equations and jump relations that are valid at the sub-scale, and that are then averaged to get equations with collective effects at the scale of interest,
- the second method postulates the form of the averaged equations, and the closure of the equations are made so as to satisfy the second principle of thermodynamics,
- the third method postulates the averaged energies of the two-phase system, and uses a variational principle to get the conservative part of the equations. Then the second principle of thermodynamics is considered to add dissipative mechanisms.

We will use the third approach, to first recover the classic models for interface capturing methods. Then, inspired by the dynamics of bubbles and by the work of Gavrilyuk and Saurel (2002), we will propose a series of enriched models, that can be used for a more general two-phase flow description and that account for some sub-scale effects. The advantage of the methodology is indeed its capacity of selecting the specific mechanisms that should be modeled. We will moreover see in Chapter 2 that our choices are relevant in the case of bubbly flows.

Throughout the derivations, we will also take care of the mathematical properties of the equations. In particular, we will be interested in *hyperbolic* systems of equations, which are esaily solved with standard numerical methods, without introducing artificial dissipation effects.

1.1 Derivation methods based on averaging procedure

1.1.1 General overview of the method

As mentioned previously, one method for deriving models that describe two-phase mixtures is based on the averaging of the continuum equations for both phases and of their relations through the interface, called *jump relations*, that are given at the sub-scale. For instance, in a bubbly flow, the sub-scale is defined as the physical scale where the characteristic lengths are smaller than the bubbles diameters. These equations and relations are then averaged, using time, space or ensemble averages (see the discussions in Ishii (1975); Drew and Passman (1999) on the choice of one of the averages techniques). This leads to averaged equations where some terms are unclosed: they can not be expressed as functions of the other variables. The final step of the modeling work consists in closing these terms. These closures usually depend on the type of two-phase flow that is modeled.

We are presenting here the broad outline of this method, and give the example of the model of Cheng, Drew, and Lahey (1985) for bubbles in a second part.

Continuum equations

The continuum equations for each phase are generally the classic balance equations for mass (let ρ be the local mass density), momentum $\rho\mathbf{u}$ (with $\mathbf{u}$ the local velocity) and energy, where ρE is the total energy defined as the sum of an internal energy ρe and the kinetic energy: $\rho E = \rho e + \frac{1}{2}\rho|\mathbf{u}|^2$. These equations are valid inside each component. They write:

$$\partial_t \rho + \mathrm{div}_{\mathbf{x}}(\rho\mathbf{u}) = 0, \tag{1.1a}$$

$$\partial_t(\rho\mathbf{u}) + \mathrm{div}_{\mathbf{x}}(\rho\mathbf{u}\mathbf{u}^T) = \mathrm{div}_{\mathbf{x}}(\mathbf{T}) + \rho\mathbf{b}, \tag{1.1b}$$

$$\partial_t(\rho E) + \mathrm{div}_{\mathbf{x}}(\rho\mathbf{u}E) = \mathrm{div}_{\mathbf{x}}(\mathbf{T}\mathbf{u} - \mathbf{q}) + \rho(r + \mathbf{b}^T\mathbf{u}), \tag{1.1c}$$

where $\mathbf{T}$ is the stress tensor, $\mathbf{b}$ contains all the body forces, $\mathbf{q}$ is a heat flux and r accounts for heating sources.

Jump relations

Due to the presence of interfaces in a two-phase mixture, balance equations are discontinuous at certain locations. One thus needs to specify relations across these interfaces, to ensure the conservation of total mass, momentum and energy. Let $\mathbf{n}$ be the interface unit normal (whose orientation is chosen arbitrarily), and $\mathbf{v}_i$ its velocity. The definition of this velocity will be discussed in Chapter 3. There exist indeed multiple ways in defining an interface (see Pope (1988)), and thus multiple possible definitions for $\mathbf{v}_i$. However, its component along the $\mathbf{n}$ vector is uniquely defined (proof in Drew and Passman (1999) and mentioned in Chapter 3).

Let us consider a piece of fluid $\mathcal{V}$ that is divided by an interface Σ_i into two sub-volumes $\mathcal{V}_1$ and $\mathcal{V}_2$ such that $\mathcal{V} = \mathcal{V}_1 \bigcup \mathcal{V}_2$. Let $\mathbf{n}_{1,i}$ and $\mathbf{n}_{2,i}$ be the outward pointing normals to Σ_i from the point of view of volume $\mathcal{V}_1$ and $\mathcal{V}_2$ respectively. Let us integrate the balance equation (such as (1.1a), (1.1b) or (1.1c)) of a quantity ϕ ($\phi \in \{\rho, \rho\mathbf{u}, \rho E\}$) over this piece of volume:

$$\int_{\mathcal{V}} \left[\partial_t \phi + \mathrm{div}_{\mathbf{x}}(\mathbf{F}) - S\right] = 0$$

$$\Leftrightarrow$$

$$\sum_{k=1,2} \int_{\mathcal{V}_k} \left[\partial_t \phi + \mathrm{div}_{\mathbf{x}}(\mathbf{F}) - S\right] = 0$$

$$\Leftrightarrow$$

$$\sum_{k=1,2} \int_{\mathcal{V}_k} \left[\partial_t \phi - S\right] + \int_{\partial\mathcal{V}_k \setminus \Sigma_i} \mathbf{F}^T \mathbf{n} + \int_{\Sigma_i} \mathbf{F}_i^T \mathbf{n}_{k,i} = 0$$

$$\Leftrightarrow$$

$$\sum_{k=1,2} \int_{\mathcal{V}_k} \left[\partial_t \phi - S\right] + \int_{\partial\mathcal{V}_k} \mathbf{F}^T \mathbf{n} - \int_{\Sigma_i} \mathbf{F}^T \mathbf{n}_{k,i} + \int_{\Sigma_i} \mathbf{F}_i^T \mathbf{n}_{k,i} = 0, \tag{1.2}$$

where S are the external effects, like forces or heat sources, and $\mathbf{F}_i$ is the definition of the flux function at the interface, involving the interfacial velocity and interfacial forces. From the balance laws (1.1a), (1.1b) and (1.1c) integrated over the volume $\mathcal{V}_k$, $k = 1, 2$, one has:

$$\int_{\mathcal{V}_k} \left[\partial_t \phi - S\right] + \int_{\partial\mathcal{V}_k} \mathbf{F}^T \mathbf{n} = \int_{\mathcal{V}_k} \left[\partial_t \phi + \mathrm{div}_{\mathbf{x}}(\mathbf{F}) - S\right] = 0.$$

Then, from (1.2) one necessarily has for any surface Σ_i:

$$\sum_{k=1,2} \int_{\Sigma_i} \mathbf{F}_i^T \mathbf{n}_{k,i} - \int_{\Sigma_i} \mathbf{F}^T \mathbf{n}_{k,i} = 0$$

$$\Leftrightarrow$$

$$\left[\mathbf{F} - \mathbf{F}_i\right]^T \mathbf{n} = 0,$$

where $[\cdot]$ denotes the jump between the two states and one may choose $\mathbf{n} = \mathbf{n}_{1,i}$.

Performing this integration for equations (1.1a), (1.1b) and (1.1c), gives the following jump relations:

$$\left[\rho(\mathbf{u} - \mathbf{v}_i)\right]^T \mathbf{n} = 0, \tag{1.3a}$$

$$\left[\rho\mathbf{u}(\mathbf{u} - \mathbf{v}_i)^T + \mathbf{T}\right] \mathbf{n} = \mathbf{m}_i^\sigma, \tag{1.3b}$$

$$\left[\rho E(\mathbf{u} - \mathbf{v}_i) + \mathbf{T}\mathbf{u} - \mathbf{q}\right]^T \mathbf{n} = \epsilon_i^\sigma, \tag{1.3c}$$

with $\mathbf{m}_i^\sigma$ denoting the surface forces and ϵ_i^σ the surface energy contribution.

When the phases are at equilibrium (there is no dynamic process at the interface), one can use the well-known Young-Laplace equation relating the jump in pressure with the surface tension effect:

$$[p] = 2\sigma H,$$

with p the dynamic pressure, H being the local mean curvature of the interface (see a definition in Chapter 3), and σ the surface tension coefficient.

Averaging step

The previous equations and jump relations are valid inside each of the component specific domain or at interfaces only. To get equations for a two-phase mixture, where each interface can not or does not need to be resolved, we need now to proceed to the averaging of the above equations and relations.

First, a characteristic function $\boldsymbol{\chi}_k$ is needed to isolate each component k during the averaging step (indeed, each component has its own physical properties such as density or viscosity):

$$\boldsymbol{\chi}_k(\mathbf{X}(t,\mathbf{x})) = \begin{cases} 1, & \text{if} \quad \mathbf{X}(t,\mathbf{x}) \text{ is in phase } k, \\ 0, & \text{else.} \end{cases}$$

Consequently, the gradient of $\boldsymbol{\chi}_k$ is related to the Dirac distribution associated with the interface $\delta_i(\mathbf{x},t)$:

$$\boldsymbol{\nabla}\boldsymbol{\chi}_k = -\delta_i(\mathbf{x},t)\mathbf{n}_{k,i}(\mathbf{x},t).$$

$\boldsymbol{\chi}_k$ follows the topological equation (Drew and Passman (1999)):

$$\partial_t\boldsymbol{\chi}_k + \mathbf{u}^T\boldsymbol{\nabla}\boldsymbol{\chi}_k = 0, \tag{1.4}$$

where the velocity field $\mathbf{u}$ corresponds to the interfacial velocity $\mathbf{v}_i$ at the interface. By multiplying equations (1.1a), (1.1b) and (1.1c) by the characteristic function $\boldsymbol{\chi}_k$, one obtains a new set of equations that writes:

$$\partial_t\left(\boldsymbol{\chi}_k\rho\right) + \operatorname{div}\left(\boldsymbol{\chi}_k\rho\mathbf{u}\right) = -\delta_i\rho\left(\mathbf{u}-\mathbf{v}_i\right)^T\mathbf{n}_{k,i}, \tag{1.5a}$$

$$\partial_t(\boldsymbol{\chi}_k\rho\mathbf{u}) + \operatorname{div}(\boldsymbol{\chi}_k\rho\mathbf{u}\mathbf{u}^T) = \operatorname{div}(\boldsymbol{\chi}_k\mathbf{T}) + \boldsymbol{\chi}_k\rho\mathbf{b} - \delta_i\left(\rho\mathbf{u}(\mathbf{u}-\mathbf{v}_i)^T - \mathbf{T}\right)\mathbf{n}_{k,i}, \tag{1.5b}$$

$$\partial_t(\boldsymbol{\chi}_k\rho E) + \operatorname{div}(\boldsymbol{\chi}_k\rho\mathbf{u}E) = \operatorname{div}(\boldsymbol{\chi}_k(\mathbf{T}\mathbf{u}-\mathbf{q})) + \boldsymbol{\chi}_k\rho(r+\mathbf{b}^T\mathbf{u}) - \delta_i\left(\rho E(\mathbf{u}-\mathbf{v}_i) - (\mathbf{T}\mathbf{u}-\mathbf{q})\right)^T\mathbf{n}_{k,i}. \tag{1.5c}$$

Let us note $\langle\cdot\rangle$ the ensemble averaging procedure. This average has the following classic properties (that is also true for other types of averages):

- linearity (λ is a constant):

$$\langle\lambda a + b\rangle = \lambda\langle a\rangle + \langle b\rangle,$$

- idempotence:

$$\langle\langle a\rangle\rangle = \langle a\rangle,$$

- Gauss and Leibniz rules:

$$\langle\partial_t a\rangle = \partial_t\langle a\rangle \quad \text{and} \quad \langle\partial_x a\rangle = \partial_x\langle a\rangle.$$

It is now possible to average the equations (1.5a), (1.5b) and (1.5c), and get equations on the following variables:

$$\begin{aligned} \alpha_k &= \langle\boldsymbol{\chi}_k\rangle, \\ m_k &= \langle\boldsymbol{\chi}_k\rho\rangle, \\ (m\mathbf{u})_k &= \langle\boldsymbol{\chi}_k\rho\mathbf{u}\rangle, \\ (mE)_k &= \langle\boldsymbol{\chi}_k\rho E\rangle. \end{aligned}$$

It is convenient to introduce also the phasic densities $\rho_k = \dfrac{m_k}{\alpha_k}$, velocities $\mathbf{u}_k = \dfrac{(m\mathbf{u})_k}{m_k}$ and energies $E_k = \dfrac{(mE)_k}{m_k}$, as well as pressures p_k from the averaging of the stress tensor $\mathbf{T}$. However, all the terms arising from the averaging process are not closed. This is especially the case of the interfacial terms, which are written in blue in equations (1.5a), (1.5b) and (1.5c), and will be noted in blue in the following. Moreover, although constitutive relations (such as the equations of states or the expression of $\mathbf{T}$) are well known in pure phases, this is no longer the case for the mixture averaged variables. New relations need to be stated, using a modeling approach that is consistent with the sub-scale structure of the flow.

Averaged equations

The set of averaged equations can finally be written ($k = 1, 2$):

$$\partial_t (\alpha_k \rho_k) + \operatorname{div} (\alpha_k \rho_k \mathbf{u}_k) = \Gamma_k, \tag{1.6a}$$

$$\partial_t(\alpha_k \rho_k \mathbf{u}_k) + \operatorname{div}(\alpha_k \rho_k \mathbf{u}_k \mathbf{u}_k^T) = \operatorname{div}(\alpha_k \mathbf{T}_k) + \alpha_k \rho_k \mathbf{b}_k + \Gamma_k \mathbf{v}_{k,i} + \mathbf{M}_k, \tag{1.6b}$$

$$\partial_t(\alpha_k \rho_k E_k) + \operatorname{div}(\alpha_k \rho_k \mathbf{u}_k E_k) = \operatorname{div}(\alpha_k(\mathbf{T}_k \mathbf{u}_k - \mathbf{q}_k)) + \alpha_k \rho_k (r_k + \mathbf{b}_k^T \mathbf{u}) + \Gamma_k E_{k,i} + \mathbf{M}_k \mathbf{v}_{k,i} + \Pi_k, \tag{1.6c}$$

where the blue terms stand for the interfacial unclosed terms.

Remark 2. *One often finds a variable named the interfacial pressure in the expression of* $\mathbf{M}_k$*, which appears then as a classic stress tensor:*

$$\mathbf{M}_k = p_{k,i} \boldsymbol{\nabla} \alpha_k + \tilde{\mathbf{M}}_k.$$

The averaging of the jump relations bring further information on the blue source terms and in particular, one has the global conservation of mass, momentum and energy given by:

$$\sum_k \Gamma_k = 0,$$

$$\sum_k \mathbf{M}_k + \mathbf{v}_{k,i} \Gamma_k = \mathbf{m}^\sigma,$$

$$\sum_k \Gamma_k E_{k,i} + \mathbf{M}_k \mathbf{v}_{k,i} + \Pi_k = \epsilon^\sigma.$$

But explicit expressions for these terms relating them with the averaged variables of system (1.6) are still to be defined, as well as an equation or a relation that enables to compute the volume fractions α_k.

For the force terms in blue and the interfacial velocity, the closure work needs to consider the sub-scale effects again. Simplifications in expressing these effects can be done through assumptions on the topology of the interface. For instance, assuming spherical bubbles enables to express a part of the jump in momentum as drag forces. Then, the expressions of these forces can be easily averaged. This ends to models that are specific to one two-phase flow configuration.

For the closure of the volume fractions, one encounters different approaches in the literature. The choices made to close these terms often determine the mathematical structures of the equations. Indeed, the averaged system of equations (1.6) is often qualified as *ill-posed*, since its convective part is not unconditionally hyperbolic. Among the closure relations, one finds:

- the equality of the phase pressures, which leads to a system that is not hyperbolic, unless velocities are also equal: $\mathbf{u}_1 = \mathbf{u}_2$ (Stewart and Wendroff (1984)),
- multiple definitions of an interfacial pressure (see for example in Stuhmiller (1977)),
- the addition of an equation on volume fraction, which helps recovering the hyperbolicity of the system.

This last choice leads to a seven-equation system similar to the Baer and Nunziato model Baer and Nunziato (1986), which is presented in the next section.

Remark 3. *In Lhuillier, Chang, and Theofanous (2013) and in Vazquez-Gonzalez (2016), a discussion is lead on the necessity of the well-posedness character of the systems and alternative solutions to the enforcing of hyperbolicity are proposed.*

1.1.2 Drew's model for bubbly flows

Let us present here the model for bubbly flows that can be found in Cheng, Drew, and Lahey (1983); Cheng, Drew, and Lahey (1985); Drew and Passman (1999) and related works. This model has been derived using an ensemble averaging procedure, with the following assumptions:

- there is no mass transfer between the gas and the liquid,
- gravity $\mathbf{g}$ acts as a bulk force,
- bubbles are spherical, and the distribution in size of the bubbles is monodisperse: all bubbles are of radius R that may evolve through time: $R(t)$.

Evolution equations for the mass of the fluids, their momentum and their enthalpy h_k, as can be found in Cheng, Drew, and Lahey (1985), write:

$$\partial_t \left(\alpha_k \rho_k\right) + \operatorname{div}\left(\alpha_k \rho_k \mathbf{u}_k\right) = 0, \tag{1.7a}$$

$$\partial_t(\alpha_k \rho_k \mathbf{u}_k) + \operatorname{div}(\alpha_k \rho_k \mathbf{u}_k \mathbf{u}_k^T) = -\alpha_k \boldsymbol{\nabla} p_k + \alpha_k \rho_k \mathbf{g} + (-1)^k \alpha_1 \left[\mathbf{F}_D + \mathbf{F}_{VM} + \mathbf{F}_R + \mathbf{F}_B\right] - \tau_k, \tag{1.7b}$$

$$\partial_t(\alpha_k \rho_k h_k) + \operatorname{div}(\alpha_k \rho_k \mathbf{u}_k h_k) = \alpha_k \left(\partial_t p_k + \mathbf{u}_k^T \boldsymbol{\nabla} p_k\right) - \tau_k \mathbf{u}_k + \frac{q''_{k,i}}{L_s}, \tag{1.7c}$$

$k = 1$ for gas and $k = 2$ for liquid.

The forces appearing in the momentum equations are:

- the drag force $\mathbf{F}_D$, given for a bubbly flow,
- the virtual mass force $\mathbf{F}_{VM}$, which is a force accounting for the acceleration of a volume of liquid, that is displaced due to the drift in velocities,
- a reaction force $\mathbf{F}_R$ due to the variations of the bubbles radii, that modifies the pressure in the liquid near bubbles,
- the Basset force $\mathbf{F}_B$.

τ_k is the shear stress at the wall (the border of the physical domain). $q''_{k,i}/L_s$ accounts for the interfacial heat transfers between the phases, due to temperature difference. All these terms get expressions that depend on the averaged variables α_k, ρ_k, $\mathbf{u}_k$, p_k and h_k Cheng, Drew, and Lahey (1983); Cheng, Drew, and Lahey (1985).

Now, the unclosed terms are the phasic pressures and the volume fraction α_1 (one has indeed $\alpha_2 = 1-\alpha_1$):

- The phasic pressures are related to the densities ρ_k and enthalpies h_k by equations of state (EOS): $p_k = p_k(\rho_k, h_k)$. These EOS also define the phase sound velocity c_k:

$$\frac{1}{c_k^2} = \left(\partial_{p_k} \rho_k\right)_{h_k} + \frac{1}{\rho_k} \left(\partial_{h_k} \rho_k\right)_{p_k}.$$

- For the gas volume fraction α_1, one uses the topological assumption on the micro-structure of the flow: the flow is indeed composed of spherical bubbles of locally unique radius $R(t)$ and with a number density $N(t, \mathbf{x})$. The gas volume fraction is then simply given by:

$$\alpha_1(t, \mathbf{x}) = \frac{4}{3} \pi R^3(t) N(t, \mathbf{x}),$$

and, with the assumption that bubbles do not coalesce nor fraction, one can assume that their number density is conserved:

$$\partial_t N + \operatorname{div}(N \mathbf{u}_1) = 0$$

Now, one has to determine an equation for $R(t)$. One can note that, by using the very strong assumption of monodispersity, an equation on the radius $R(t)$ can be determined by considering the physics of a single bubble. In particular, in the following, we are interested in pulsating bubbles. In Cheng, Drew, and Lahey (1983); Cheng, Drew, and Lahey (1985), the evolution of $R(t)$ for a bubble that pulsates with a frequency ω is given by an equation close to the Rayleigh-Plesset equation, that reads:

$$p_{1,i} - p_2 = \frac{\rho_2 R \ddot{R}}{1 - ik(\omega)R} + \rho_2 \dot{R}^2 \left[\frac{2}{1 - ik(\omega)R} - \frac{1}{2} - \left(\frac{k(\omega)R}{1 - ik(\omega)R}\right)^2\right] + 4\frac{\mu_2 \dot{R}}{R} \left[1 - \frac{(k(\omega)R)^2}{2(1 - ik(\omega)R)}\right] + \frac{2\sigma}{R},$$

(1.8)

where $k(\omega)$ is the wavelength associated with pulsation ω, $p_{1,i}$ is the gas pressure at the bubble interface and μ_2 is the liquid dynamic viscosity. The notation $\dot{-}$ indicates the temporal derivative: $\dot{R} = \frac{\mathrm{d}R}{\mathrm{d}t}$. One can note that, up to now, there is no relation for p_{1i}.

Remark 4. *The expression of the interfacial heat transfer term, $q''_{k,i}/L_s$, also requires the definition of an interfacial temperature $T_{1,i}$, that must be determined in the same way as $p_{1,i}$.*

From Cheng, Drew, and Lahey (1983); Cheng, Drew, and Lahey (1985), there is no closure for the interfacial variables. Only complex functions enable to relate them to the radius R and the frequency of pulsation ω in a linearized version of the equations. This linearization is further discussed in Chapter 2.

1.1.3 Conclusion on the averaging procedure

We have presented one of the ways in which averaged equations for two-phase flows can be derived. This first method, based on averaging equations that describe the sub-scale physics, enables to get systems of equations containing plenty of terms standing for averaged physical effects. But these terms require also an important modeling effort, as well as the closure of all terms of the equations. These closures are usually based on the sub-scale topology of the interface (for example spherical bubbles). Moreover, the complexity of the formulae used for these averaged forces or energies makes it usually hard to discriminate prevailing from neglected physical effects, and to simplify the equations according to different flow regimes. It is consequently difficult to get the structure of the equations that would result from simplifying assumptions (such as mechanical equilibrium). Finally, the mathematical structure of these systems of equations is in general not well-known: they are often not hyperbolic, and their dissipative structures are often not well identified.

1.2 A method based on postulated equations and second principle of thermodynamics

A second method to derive two-fluid systems of equations consists in postulating equations according to the *continuum mixture* theory (Truesdell (1969)). With this approach, equations of conservation are first given for each phase. These equations also comprise source and exchange terms that are to be closed. This closure is performed by considering thermodynamics laws for each phase and ensuring the second principle of thermodynamics for the two-phase mixture. The choices made during this closure step are not unique, as noted in Bdzil, Menikoff, Son, Kapila, and Stewart (1999), and the parameters they involve must be physically identified. We detail below this method, as it was first used in Baer and Nunziato (1986) for two-phase granular flows. The resulting model is known as the Baer and Nunziato (BN) model.

Postulated equations

The equations for the two-phase system are assumed to be given for the densities ρ_k, velocities $\mathbf{u}_k$ and internal energies e_k by ($k = 1, 2$):

$$\partial_t\left(\alpha_k\rho_k\right) + \mathbf{u}_k^T\boldsymbol{\nabla}(\alpha_k\rho_k) = -\alpha_k\rho_k\mathrm{div}\left(\mathbf{u}_k\right) + \Gamma_k, \tag{1.9a}$$

$$\alpha_k\rho_k\partial_t\mathbf{u}_k + \alpha_k\rho_k\mathbf{u}_k^T\boldsymbol{\nabla}\mathbf{u}_k + \boldsymbol{\nabla}\pi_k = \alpha_k\rho_k\mathbf{b}_k - \Gamma_k\mathbf{u}_k + \mathbf{M}_k, \tag{1.9b}$$

$$\alpha_k\rho_k\partial_t e_k + \alpha_k\rho_k\mathbf{u}_k^T\boldsymbol{\nabla}e_k = -\pi_k\mathrm{div}(\mathbf{u}_k) + \alpha_k\rho_k r_k - \mathrm{div}(\mathbf{q}_k) - \Gamma_k\left(e_k - \frac{1}{2}\mathbf{u}_k^T\mathbf{u}_k\right) - \mathbf{M}_k\mathbf{u}_k + E_k, \tag{1.9c}$$

where Γ_k, $\mathbf{M}_k$ and E_k are respectively the mass, momentum and energy exchange terms, $\mathbf{b}_k$, r_k and $\mathbf{q}_k$ are the bulk forces, the external heat sources and the heat fluxes, and π_k are the phase pressures. All these terms must be given constitutive laws and one should also derive an equation for one of the volume

fractions α_k (either a closure law, or an evolution law). The exchange terms are first constrained by the so-called *summing rule*, that ensures the conservation of the mixture quantities:

$$\sum_k \Gamma_k = 0, \quad \sum_k \mathbf{M}_k = 0, \quad \sum_k E_k = 0.$$

The second law of thermodynamics, as presented below, will provide further constraints on these terms.

Entropy function and inequality

Following the lines of Truesdell (1969), to each phase is associated a thermodynamical entropy η_k and an entropy flux:

$$\mathbf{G}_k = \alpha_k \rho_k \eta_k \mathbf{u}_k + \frac{\mathbf{q}_k - \alpha_k \rho_k r_k}{T_k},$$

with T_k the phase temperature. A priori, the entropy η_k is a function of all the phase variables: $\eta_k(\alpha_k, \rho_k, \mathbf{u}_k, T_k)$, and one of the novelty of the work of Baer and Nunziato (1986) is to consider the volume fraction α_k as an independent variable.

The second law of thermodynamics for the two-phase flow states that the following entropy inequality must be satisfied:

$$\sum_k \partial_t(\alpha_k \rho_k \eta_k) + \operatorname{div}(\mathbf{G}_k) \geqslant 0. \tag{1.10}$$

The procedure consists now in expanding the inequality (1.10), by using the state laws that relate the entropy η_k with the density ρ_k, the pressure p_k and the temperature T_k, and to make choices so that the *expanded* inequality is always satisfied. We will not detail this procedure here, since a similar approach will be presented in the next section in a simplified context. Let us just give the resulting relations that have been derived in Baer and Nunziato (1986).

Closure choices of the BN model

The choices that have been made to close the equations (1.9a), (1.9b) and (1.9c), and to satisfy the entropy inequality (1.10) provide expressions for the exchange terms that make appear three *relaxation* processes:

- a relaxation for the mass exchanges, that depends on the pressure of the solid phase and the difference between the thermodynamical states of the two phases. This effect is driven by a coefficient ϵ,
- a relaxation for the momentum exchanges under the form of drag forces, driven by a coefficient δ,
- relaxation effects in the transfer of energy, driven by a heat coefficient h.

The entropy inequality also enables to set an evolution law for the volume fractions. Let us consider here α_1 (α_2 is given by $\alpha_1 + \alpha_2 = 1$), its equation writes:

$$\partial_t \alpha_1 + \mathbf{u}_1^T \boldsymbol{\nabla} \alpha_1 = \frac{\alpha_1 \alpha_2}{\mu} \left(p_1 - (p_2 + \beta_1)\right) + \frac{\Gamma_1}{\rho_1}, \tag{1.11}$$

with β_1 a *configuration pressure*, that represents the contact forces between the solid grains in the context of granular beds and μ a *compaction viscosity*.

Now, the last step of the procedure consists in providing the different coefficients a physical interpretation and expressions. Therefore, this step depends on the type of two-phase flow that is modeled. In Baer and Nunziato (1986), expressions for ϵ, δ and h were found in the literature, as these phenomena had already been studied experimentally. However, μ was a new parameter, whose effect was not known at the time of the article. This parameter has since appeared in many two-phase flows models, and has been identified with viscous effects (see in Gavrilyuk and Saurel (2002) for instance). We present in Chapter 2 a new identification for this parameter.

Conclusion

The model of Baer and Nunziato has been the subject of multiple studies and is widely used in the granular media community (see Gallouët, Helluy, Hérard, and Nussbaum (2010) for example and a review in Bdzil, Menikoff, Son, Kapila, and Stewart (1999)).

Comparing with the method of Section 1.1, the approach that was presented here has the advantage of better controlling the dissipative effects of the equations, due to their compliance with the second principle of thermodynamics. Moreover, the seven-equation system is shown to be hyperbolic Embid and Baer (1992). However, the methodology introduces parameters whose physical meaning is to be found a posteriori.

The relaxation parameters can be used to derive reduced-order models, as was proposed in Kapila, Menikoff, Bdzil, Son, and Stewart (2001). In this paper, the characteristic times for velocity and pressure relaxations are studied, and successively assumed to be negligible in comparison with the other characteristic times of the system. A model with unique velocity and then a model with unique pressure are thus derived from the BN model. The idea of a hierarchy of models will be studied at the end of this chapter for a series of other models, but that are also related through relaxation effects.

1.3 A method based on a variational principle and entropy dissipation

Now, let us present the last approach that can be used for the derivation of two-phase models. It has similarities with the method of section 1.2, since it also involves the second principle of thermodynamics. However, the convective part of the equations is firstly obtained by a variational principle, after having postulated the energies for the two-phase system. After a few words on the origins and the history of the **Least Action Principle**, that may help understand its theoretical grounds, we will use it to derive mixture models for gas-liquid two-phase flows.

1.3.1 Origins of the variational principle and general overview

Origins and a history of the Least Action Principle are presented in Berdichevsky (2009): the idea that *Nature* tries to minimize its *efforts* dates back to Aristotle, but first concrete equations were written by Fermat for solving the problem of light refraction at the interface of two different media. At that time, it was believed that light takes the path that minimizes its time of propagation from one point to the other, and so was it for every mechanical process. It evolved to the belief that Nature tries in fact to minimize its effort (not only in terms of time). Maupertuis introduced the notion of *action*, which, for a solid object, is the product of its mass by its velocity and by the length of the considered path. This definition of the action was improved by Euler. From his works on differential calculus, Euler used instead the integral of Maupertuis' action over the path, which is equivalent to the integration over time of the kinetic energy of the system. Later, Lagrange considered another form of the action functional, and postulated that the true motion is given by a stationary point of this action under the constraint of energy conservation. This formulation lead to the well-known Lagrange's equations:

$$\frac{\partial \mathcal{L}}{\partial q_i} - \frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{q}_i} = 0,$$

where q_i are N generalized coordinates that describe the system, $\dot{q}_i$ are their derivatives with time and $\mathcal{L}$ is the Lagrangian functional defined as the difference between kinetic and potential energies:

$$\mathcal{L} = E_{kin} - E_{pot}. \tag{1.12}$$

Finally, Hamilton defined the action as the integral over time of the Lagrangian functional, from t_0 where the system lies at state q_0, to a time t_1 with a system at state q_1:

$$\mathcal{A} = \int_{t_0}^{t_1} \mathcal{L}(q, \dot{q}, t) dt. \tag{1.13}$$

Hamilton's variational principle states that (see Berdichevsky (2009)): *"The true motion is the stationary point of the functional* (1.13) *on the set of all paths beginning at point* q_0 *and instant* t_0 *and ending at point* q_1 *at instant* t_1*."* This formulation of the variational principle, as well as the hamiltonian action are currently one of the two most used forms (with Jacobi's one).

Remark 5. *One can note that, according to the Least Action Principle, two contributions seem to rule the system evolution: the kinetic energy and potential energies. In the usual equations of mechanics, these two contributions are also linked as, for example, in the second law of dynamics:*

$$m\frac{\mathrm{d}v}{\mathrm{d}t} = F,$$

with F *the forces that are applied to the system. In this equation, the forces are not all issued from potential energies however: for instance, one can consider friction effects, that are dissipative forces. These dissipative effects cannot be deduced from the Least Action Principle, they must be added afterwards by other means.*

This historical approach mainly concerned mechanics of solid systems. Using the variational principle to derive equations for continuous systems has then developed and in particular, barotropic two-phase flow systems were derived with this method in Berdichevsky (1983); Berdichevsky (2009); Bedford and Drumheller (1983). A methodology for deriving incompressible two-phase equations is given in Geurst (1986) and a first comparison between these equations and a model issued from the averaging procedure was done in Pauchon and Smereka (1992). All these works used Lagrange multipliers to solve the optimization problem that is represented by the Least Action Principle, under the constraints of mass conservation. Later, these constraints were formulated in the form of *virtual displacements* in Lagrangian coordinates, that was simplifying the procedure (see Gavrilyuk, Gouin, and Perepechko (1998); Gavrilyuk and Gouin (1999); Gavrilyuk and Saurel (2002)). We will use the same techniques in the following paragraphs.

1.3.2 Example of application of the variational principle in the case of a gas-liquid two-phase flow

1.3.2.1 Main assumptions and definition of the variables

To illustrate the methodology, as proposed in Gavrilyuk and Saurel (2002), we apply it to derive the 3-equation homogeneous equilibrium barotropic two-phase model (HEM), for mixtures of gas and liquid, from Chanteperdrix, Villedieu, and Vila (2002). The derivations are indeed quite simple for this model, since it is based on the following assumptions:

(H1) *homogeneity*: at each point of the mixture, gas and liquid share the same velocity $\mathbf{u}$,

(H2) *equilibrium*: gas and liquid stand at mechanical equilibrium ($p_{gas} = p_{liq}$),

(H3) *barotropic*: the thermodynamical state does not depend on temperature (the system is either isothermal or isentropic),

(H4) there is no phase transition, thus the mass of each component is conserved.

We will see in Chapter 2 how these assumptions can be justified in the context of bubbly media. We will also see that by assuming (H1), assumption (H2) is a natural consequence of the least action principle.

Let us consider a gas-liquid mixture and note ρ_k, $k = 1, 2$ the mass densities of both components. In the following, we will use the index 1 for gas and 2 for liquid.

Equations of state

Following hypothesis (H3), we use barotropic Equations of State (EOS):

$$\rho_k \mapsto f_k(\rho_k) \quad \text{and} \quad \rho_k \mapsto p_k(\rho_k) = \rho_k^2(\mathrm{d}f_k/\mathrm{d}\rho_k)(\rho_k), \tag{1.14}$$

where f_k and p_k are respectively the specific Helmholtz free energies and the partial pressures of each fluid, $k = 1, 2$. $c_k^2 = (\mathrm{d}p_k/\mathrm{d}\rho_k)$ denotes the sound velocities within each pure material k.

Mass conservation

Let us define m_1 and m_2 as the mass concentrations of gas and liquid respectively: they are the masses of these components per unit volume. Following assumption (H4), we have the two following conservation equations for m_1 and m_2:

$$\partial_t m_1 + \mathrm{div}(m_1 \mathbf{u}) = 0, \tag{1.15}$$

$$\partial_t m_2 + \mathrm{div}(m_2 \mathbf{u}) = 0. \tag{1.16}$$

Then the total density of the mixture, $\rho = m_1 + m_2$ is also conserved due to assumption (H1):

$$\partial_t \rho + \mathrm{div}(\rho \mathbf{u}) = 0. \tag{1.17}$$

Let us define the mass fractions of gas $Y_1 = m_1/\rho$ and of liquid $Y_2 = m_2/\rho$. In the following, we note $Y = Y_1$. Consequently, we also have $Y_2 = 1 - Y$, and from (1.15), one can deduce directly that the conservation equation for Y writes:

$$\partial_t (\rho Y) + \mathrm{div}(\rho Y \mathbf{u}) = 0. \tag{1.18}$$

Volume fractions

The volume fractions of the components, α_1 and α_2, are defined as the ratio of their volume occupation over the control volume, and their sum is equal to 1 for immiscible components. By definition, we have $\alpha_1 = m_1/\rho_1$ and $\alpha_2 = m_2/\rho_2$. This definition also shows how the volume fractions are related to the components thermodynamical state. This will be useful later. In the following, we note $\alpha = \alpha_1$ and then $\alpha_2 = 1 - \alpha$.

Let us outline here that all the variables defined above are averaged variables for the mixture.

1.3.2.2 Derivation of conservative equations using the variational principle

The first step in the modeling methodology aims at deriving a system of conservation laws, that is hyperbolic and equipped with a mathematical entropy evolution equation. Following the lines of Berdichevsky (2009) for barotropic fluids, of Gavrilyuk and Saurel (2002) for a system of equations for two compressible fluids and two temperatures, or of Caro, Coquel, Jamet, and Kokh (2006) for a homogeneous isothermal two-fluid model, we propose to use a variational approach and Hamilton's principle of stationary action to derive the conservative structure of the two-fluid model satisfying the assumptions (H1) to (H4).

Set of variables

Among the variables that can describe the mixture, we choose in this part to use the set $(\rho, Y, \mathbf{u}, \alpha)$.

Energies and Lagrangian

The first and most difficult step of the method is the definition of the system energies, in terms of the averaged variables of interest. Here we only present the method and we will not discuss the possible choices for the expression of these energies. This is done in the next sections. Let us just assume that the bulk kinetic energy for the mixture writes:

$$E_{kin} = \frac{1}{2}\rho |\mathbf{u}|^2,$$

and the potential energies are given by Helmholtz's free energy for the mixture (defined later):

$$E_{pot} = \rho f(\rho, Y, \alpha).$$

The Lagrangian $\mathcal{L} = E_{\mathrm{kin}} - E_{\mathrm{pot}}$ of our system is fully defined, namely:

$$\mathcal{L}(\mathbf{u}, \rho, Y, \alpha) = \frac{1}{2}\rho |\mathbf{u}|^2 - \rho f(\rho, y, \alpha). \tag{1.19}$$

Lagrangian description

Let $\mathscr{V}(t)$ be the volume occupied by a portion of fluid during $t_0 \leqslant t \leqslant t_1$ (see Figure 1.1). We note

$$\Omega = \{(\mathbf{x}, t) \in \mathbb{R}^3 \times [t_0, t_1] \mid \mathbf{x} \in \mathscr{V}(t)\},$$

the subset of all space-time points crossed by $\mathscr{V}(t)$ for $t_0 \leqslant t \leqslant t_1$.

The space variable $\mathbf{X} \in \mathscr{V}(0)$ denotes the Lagrangian coordinates associated with the reference frame at the initial instant $t = t_0$. If $(\mathbf{X}, t) \mapsto \boldsymbol{\varphi}$ is the mapping that gives the position $\boldsymbol{\varphi}(\mathbf{X}, t)$ at instant t of a fluid element that was located at $\mathbf{X}$ at time $t = t_0$, then obviously

$$\Omega = \{(\boldsymbol{\varphi}(\mathbf{X}, t), t) \mid \mathbf{X} \in \mathscr{V}(t_0),\ t_0 \leqslant t \leqslant t_1\}.$$

Eulerian equivalence

From a pure Eulerian point of view, a transformation of the medium is fully characterized by the fields $(\mathbf{x}, t) \mapsto (\rho, \mathbf{u}, Y, \alpha)$. Equivalently, we can say that a transformation of the medium is fully characterized by the Eulerian fields $(\mathbf{x}, t) \mapsto (Y, \alpha)$ and the Lagrangian mapping $(\mathbf{X}, t) \mapsto \boldsymbol{\varphi}$ under the hypothesis that $\boldsymbol{\varphi}$ is compliant with the mass conservation.

Family of path

If $(\mathbf{x}, t) \mapsto (Y, \alpha)$ and $(\mathbf{X}, t) \mapsto \boldsymbol{\varphi}$ is a given transformation of the medium, we consider a family of transformations $(\mathbf{x}, t, \zeta) \mapsto (\widehat{Y}, \widehat{\alpha})$ and $(\mathbf{X}, t, \zeta) \mapsto \widehat{\boldsymbol{\varphi}}$ parametrized by $\zeta \in [0, 1]$ such that:

- the **physical path** is obtained when $\zeta = 0$:

$$(\widehat{Y}, \widehat{\alpha})(\mathbf{x}, t, \zeta = 0) = (Y, \alpha)(\mathbf{x}, t) \quad \text{and} \quad \widehat{\boldsymbol{\varphi}}(\mathbf{X}, t, \zeta = 0) = \boldsymbol{\varphi}(\mathbf{X}, t), \tag{1.20}$$

- $\widehat{Y}$ and $\widehat{\boldsymbol{\varphi}}$ verify the mass and partial mass conservation equations (1.17) and (1.18), for all $\zeta \in [0, 1]$.

This family of transformations defines all the conceivable paths from $\mathscr{V}(t_0)$ to $\mathscr{V}(t_1)$ that the variables of the system, and consequently the Lagrangian functional, could follow (see an illustration in Figure 1.1).

Hamilton's action

The Hamiltonian Action $\mathcal{A}(\zeta)$ for the piece of fluid between times t_0 and t_1 writes:

$$\mathcal{A}(\zeta) = \int_\Omega \mathcal{L}(\widehat{\mathbf{u}}, \widehat{\rho}, \widehat{Y}, \widehat{\alpha})\, \mathrm{d}\mathbf{x}\mathrm{d}t.$$

The Least Action Principle will then provide the physical path as the one minimizing the variations of Hamilton's action.

Infinitesimal transformation

Now, to explicitly get these minimal variations, we adopt the classic definition of the infinitesimal transformations that act on the medium by introducing the infinitesimal displacement $(\mathbf{x}, t) \mapsto \boldsymbol{\xi}$ around the *physical* path, where

$$\boldsymbol{\xi}\big(\boldsymbol{\varphi}(\mathbf{X}, t), t\big) = \left(\frac{\partial \widehat{\boldsymbol{\varphi}}}{\partial \zeta}\right)_{\mathbf{X}, t} (\mathbf{X}, t, \zeta = 0),$$

and by setting for any Eulerian field $(\mathbf{x}, t, \zeta) \mapsto \widehat{b}$ the **infinitesimal variation**:

$$\delta b(\mathbf{x}, t) = \left(\frac{\partial \widehat{b}}{\partial \zeta}\right)_{\mathbf{x}, t} (\mathbf{x}, t, \zeta = 0). \tag{1.21}$$

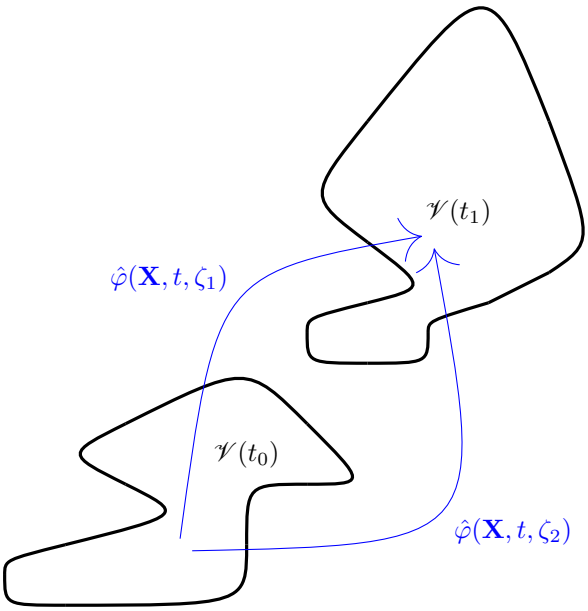


Figure 1.1: *Scheme representing two different possible paths $\hat{\phi}$ parametrized by ζ_1 and ζ_2, that transform the piece of fluid $\mathcal{V}$ from it state at time t_0 to its state at time t_1.*

Watching along the lines of Berdichevsky (2009); Gavrilyuk and Saurel (2002); Gavrilyuk (2012), our hypotheses provide now the infinitesimal variations of ρ, Y and $\mathbf{u}$ around the physical path. We obtain indeed the following relations between these infinitesimal variations and the infinitesimal displacement:

$$\delta\rho = -\mathrm{div}(\rho\boldsymbol{\xi}), \quad \delta\mathbf{u} = D_t\boldsymbol{\xi} - (\boldsymbol{\xi}^T\boldsymbol{\nabla})\mathbf{u} \quad \text{and} \quad \delta Y = -(\boldsymbol{\xi}^T\boldsymbol{\nabla})Y. \tag{1.22}$$

Remark 6. *The details of the derivation of these relations are not given here, as they involve tedious computations. However, one can note that one of way to get them is to switch the roles of t and ζ and derive the equations (1.22) like conservation equations.*

Let us also note that here the infinitesimal variations are given for $\mathcal{A}$ by:

$$\delta\mathcal{A} = \frac{\mathrm{d}\mathcal{A}}{\mathrm{d}\zeta}(\zeta = 0).$$

To express the stationary state, let us now compute the infinitesimal variations of the action:

$$\delta\mathcal{A} = \int_\Omega \left[\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right)^T \delta\mathbf{u} + \frac{\partial\mathcal{L}}{\partial\rho}\delta\rho + \frac{\partial\mathcal{L}}{\partial Y}\delta Y + \frac{\partial\mathcal{L}}{\partial\alpha}\delta\alpha \right] (\mathbf{u}, \rho, Y, \alpha)\, \mathrm{d}\mathbf{x}\mathrm{d}t.$$

We make the classic assumption (see Gavrilyuk and Gouin (1999); Gavrilyuk and Saurel (2002)) that for our transformation family, $\boldsymbol{\xi}$ and $\delta\alpha$ vanish on $\partial\Omega$. The Green formula and the relations (1.22) allow us to write

$$\begin{aligned}\delta\mathcal{A} = &-\int_\Omega \left\{ -\rho\boldsymbol{\nabla}\left(\frac{\partial\mathcal{L}}{\partial\rho}\right) + \partial_t\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right) + \mathrm{div}\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\mathbf{u}^T\right) + (\boldsymbol{\nabla}\mathbf{u})^T\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right) + \left(\frac{\partial\mathcal{L}}{\partial Y}\right)\boldsymbol{\nabla}Y \right\}^T \boldsymbol{\xi}\, \mathrm{d}\mathbf{x}\mathrm{d}t \\ &+ \int_\Omega \frac{\partial\mathcal{L}}{\partial\alpha}\delta\alpha\, \mathrm{d}\mathbf{x}\mathrm{d}t.\end{aligned} \tag{1.23}$$

Least Action Principle

Finally the Least Action Principle boils down to postulate that a physical transformation of the medium should extremize the Hamiltonian action $\mathcal{A}$:

$$\delta\mathcal{A} = 0.$$

Both parts of the integral (1.23) should satisfy this equality, whatever the choices of ξ and $\delta\alpha$. This yields

$$-\rho\boldsymbol{\nabla}\left(\frac{\partial\mathcal{L}}{\partial\rho}\right)+\partial_t\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right)+\operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\mathbf{u}^T\right)+(\boldsymbol{\nabla}\mathbf{u})^T\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right)+\left(\frac{\partial\mathcal{L}}{\partial Y}\right)\boldsymbol{\nabla}Y = \mathbf{0}, \tag{1.24}$$

$$\frac{\partial\mathcal{L}}{\partial\alpha} = 0. \tag{1.25}$$

For the choice of $\mathcal{L}$ expressed by (1.19), we have

$$\frac{\partial\mathcal{L}}{\partial\rho}=\frac{|\mathbf{u}|^2}{2}-f-\rho\frac{\partial f}{\partial\rho}, \qquad \frac{\partial\mathcal{L}}{\partial Y}=-\rho\frac{\partial f}{\partial Y}, \qquad \frac{\partial\mathcal{L}}{\partial\alpha}=-\rho\frac{\partial f}{\partial\alpha}, \qquad \frac{\partial\mathcal{L}}{\partial\mathbf{u}}=\rho\mathbf{u}. \tag{1.26}$$

Conservative equations

Relations (1.24) and (1.25) will respectively provide the evolution equations for the momentum and closure relation for the volume fraction. Indeed, reinjecting (1.26) into (1.24) provides

$$\partial_t(\rho\mathbf{u})+\operatorname{div}(\rho\mathbf{u}\mathbf{u}^T)+\boldsymbol{\nabla}\left(\rho^2\frac{\partial f}{\partial\rho}\right)=\mathbf{0}.$$

Then, we define the medium pressure p by setting

$$p=\rho^2\frac{\partial f}{\partial\rho}. \tag{1.27}$$

Using the mass and partial mass conservation hypotheses we obtain that the fluid transformations are governed by the following system of equations

$$\partial_t\rho+\operatorname{div}(\rho\mathbf{u})=0, \tag{1.28a}$$

$$\partial_t(\rho Y)+\operatorname{div}(\rho Y\mathbf{u})=0, \tag{1.28b}$$

$$\partial_t(\rho\mathbf{u})+\operatorname{div}(\rho\mathbf{u}\mathbf{u}^T)+\boldsymbol{\nabla}p=\mathbf{0}, \tag{1.28c}$$

$$\rho\frac{\partial f}{\partial\alpha}=0. \tag{1.28d}$$

We recall here that the Least Action Principle only provides the conservative part of our models. More specifically, it enables to derive the momentum equation (1.28c), that supplements the mass conservation equations (1.28a)-(1.28b) which have been postulated. Let us also note that the last equation (1.28d) is a priori not an evolution law, but rather a closure relation, related to the choice of the thermodynamics laws.

Choice of a mixture free energy

For instance, if we specify the free energy of the medium as the bulk mixture free energy, we set

$$f(\rho,Y,\alpha)=Yf_1\left(\frac{\rho Y}{\alpha}\right)+(1-Y)f_2\left(\frac{\rho(1-Y)}{1-\alpha}\right). \tag{1.29}$$

For this choice, granted that $\rho_k^2\partial f_k/\partial\rho_k=p_k$, $k=1,2$, a straightforward calculation gives that

$$p=\rho^2\frac{\partial f}{\partial\rho}=\alpha p_1+(1-\alpha)p_2 \quad \text{and} \quad \rho\frac{\partial f}{\partial\alpha}=p_2-p_1.$$

and system (1.28) can then be written:

$$\partial_t\rho+\operatorname{div}(\rho\mathbf{u})=0, \tag{1.30a}$$

$$\partial_t(\rho Y)+\operatorname{div}(\rho Y\mathbf{u})=0, \tag{1.30b}$$

$$\partial_t(\rho\mathbf{u})+\operatorname{div}(\rho\mathbf{u}\mathbf{u}^T)+\boldsymbol{\nabla}p=\mathbf{0}, \tag{1.30c}$$

$$p_1=p_2. \tag{1.30d}$$

We can see that for this choice of free energy, the Least Action Principle naturally leads to a system with the equilibrium of the phase pressures. This 3-equation system is well-known in the two-fluid community: see for instance an application to sloshing in Chanteperdrix, Villedieu, and Vila (2002), where its mathematical properties are presented. We recall some of these properties briefly below.

Mathematical properties

[P1] The equations (1.30a) to (1.30d) define a closed system. Indeed, ρ, Y and $\mathbf{u}$ are associated to their evolution equations. p_1 and p_2 are computed by using their barotropic pressure laws:

$$p_1(\rho_1) = p_1\left(\frac{\rho Y}{\alpha}\right) \quad \text{and} \quad p_2(\rho_2) = p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right).$$

Finally, one can compute the gas volume fraction α by using the barotropic laws and relation (1.30d). In Chanteperdrix, Villedieu, and Vila (2002), it is shown that for linearized pressure laws, the equilibrium volume fraction α is the unique root of a second degree polynomial equation that lies in $[0,1]$.

[P2] In one dimension, system (1.30) is hyperbolic, with three eigenvalues that are u, $u - \mathrm{c}_{\mathrm{Wood}}$ and $u + \mathrm{c}_{\mathrm{Wood}}$. $\mathrm{c}_{\mathrm{Wood}}$ is Wood's sound velocity for two-phase mixtures with pressure equilibrium (Wood (1930)). It is defined by:

$$\mathrm{c}_{\mathrm{Wood}} = \left[\rho\left(\frac{\alpha}{\rho_1 c_1^2} + \frac{1-\alpha}{\rho_2 c_2^2}\right)\right]^{-1/2}. \tag{1.31}$$

For fixed values of ρ_1, ρ_2, c_1 and c_2, $\mathrm{c}_{\mathrm{Wood}}$ is a convex function of α over $[0,1]$. What is more, for a large portion of the segment $[0,1]$, the value of $\mathrm{c}_{\mathrm{Wood}}$ is smaller (and even largely smaller) than the values of c_1 and c_2 (see in Figure 1.2 Wood's velocity given as a function of volume fraction).

[P3] A mathematical entropy for system (1.30) is given by (see Godlewski and Raviart (1996); Serre (2007)):

$$\eta(\mathbf{u}, \rho, Y, \alpha) = \frac{1}{2}|\mathbf{u}|^2 + f(\rho, Y, \alpha). \tag{1.32}$$

One can note that this entropy is in fact equal to the total energy of the system and is then different from the usual thermodynamical entropy. As explained in Serre (2007), this expression for entropy is obtained by noting that Euler's isothermal system of equation can be considered as the asymptotic limit of Euler-Fourier system (Euler equations with a Fourier's dissipative source term). Then by using relations that exist then between the entropies of these two systems, one recovers the entropy function (1.32). The dissipative structures of system (1.28) associated to the entropy inequality are examined in the next section.

1.3.2.3 Inclusion of dissipative effects using the second principle of thermodynamics

The aim of this section is to show how to provide a conservative system, obtained through the Least Action Principle for instance, with dissipative effects that satisfy the second principle of thermodynamics, that states that the evolution equation of the entropy (1.32) must be signed: increasing physical entropy or decreasing mathematical entropy. The difference between both entropies is illustrated in the case of the Euler system for gas dynamics, for which, if the physical entropy is denoted s, then the mathematical entropy is defined as $\eta = -\rho s$, which is a convex function of the variables of the system. For both of these definitions, the entropy variations denote the passage of energy from the scale of resolution of the equations to the sub-scales. More specifically, in this section, we aim at describing small scale dissipation at the fluids interface. This way, the damping of local sub-scale interfacial mechanisms due to various dissipative phenomena can be taken into account. This approach is the same as in section 1.2. Here, it is developed in more detail in a simplified context.

First, let us suppose that equations (1.30a)-(1.30c) are valid, but we discard the pressure equality closure (1.30d) and consider again that α is an independent variable (in the same way as in the definition of the Lagrangian functional).

We now seek for an entropy flux function $\mathbf{G}$ and a proper evolution principle for $D_t\alpha$, such that

$$\partial_t(\rho\eta) + \operatorname{div}(\rho\eta\mathbf{u} + \mathbf{G}) \leqslant 0,$$

or equivalently

$$\rho D_t\eta + \operatorname{div}(\mathbf{G}) \leqslant 0, \tag{1.33}$$

where the notation $D_t\cdot = \partial_t \cdot + \mathbf{u}^T\boldsymbol{\nabla}\cdot$ is the material derivative. Relation (1.33) reads

$$\operatorname{div}(\mathbf{G}) + \rho\left(\frac{\partial\eta}{\partial\mathbf{u}}\right)^T D_t\mathbf{u} + \rho\frac{\partial\eta}{\partial\rho}D_t\rho + \rho\frac{\partial\eta}{\partial Y}D_tY + \rho\frac{\partial\eta}{\partial\alpha}D_t\alpha \leqslant 0.$$

Using (1.30a)-(1.30c) to express $D_t\mathbf{u}$, $D_t\rho$, D_tY and the relations:

$$\frac{\partial\eta}{\partial\mathbf{u}} = \mathbf{u}, \qquad \frac{\partial\eta}{\partial\alpha} = \frac{\partial f}{\partial\alpha}, \qquad \frac{\partial\eta}{\partial\rho} = \frac{\partial f}{\partial\rho},$$

we obtain that

$$\operatorname{div}\left[\mathbf{G} - p\mathbf{u}\right] + \rho\frac{\partial f}{\partial\alpha}D_t\alpha \leqslant 0. \tag{1.34}$$

A simple choice for ensuring (1.34) consists in setting

$$\mathbf{G} = p\mathbf{u} \quad \text{and} \quad \frac{\partial f}{\partial\alpha} = -\epsilon D_t\alpha,$$

where $\epsilon > 0$ is a constant. This yields a definition of $\mathcal{G}$ and an evolution equation for α that reads

$$\partial_t\alpha + \mathbf{u}^T\boldsymbol{\nabla}\alpha = -\frac{1}{\epsilon}\frac{\partial f}{\partial\alpha}.$$

Remark 7. *The choices that have been made previously are arbitrary. We could also have chosen:*

$$\frac{\partial f}{\partial\alpha} = 0$$

to satisfy the entropy inequality (corresponding to $\epsilon = 0$). In this case, the free energy is independent of α and we recover the non-dissipative system (1.30).

Now, the dissipative system of equations reads:

$$\partial_t\rho + \operatorname{div}(\rho\mathbf{u}) = 0, \tag{1.35a}$$
$$\partial_t(\rho Y) + \operatorname{div}(\rho Y\mathbf{u}) = 0, \tag{1.35b}$$
$$\partial_t(\rho\mathbf{u}) + \operatorname{div}(\rho\mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla}p = \mathbf{0}, \tag{1.35c}$$
$$\partial_t\alpha + \mathbf{u}^T\boldsymbol{\nabla}\alpha = -\frac{1}{\epsilon}\frac{\partial f}{\partial\alpha}. \tag{1.35d}$$

For the choice of the form of free energy (1.29), system (1.35) finally reads:

$$\partial_t\rho + \operatorname{div}(\rho\mathbf{u}) = 0, \tag{1.36a}$$
$$\partial_t(\rho Y) + \operatorname{div}(\rho Y\mathbf{u}) = 0, \tag{1.36b}$$
$$\partial_t(\rho\mathbf{u}) + \operatorname{div}(\rho\mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla}p = \mathbf{0}, \tag{1.36c}$$
$$\partial_t\alpha + \mathbf{u}^T\boldsymbol{\nabla}\alpha = \frac{p_1 - p_2}{\epsilon}. \tag{1.36d}$$

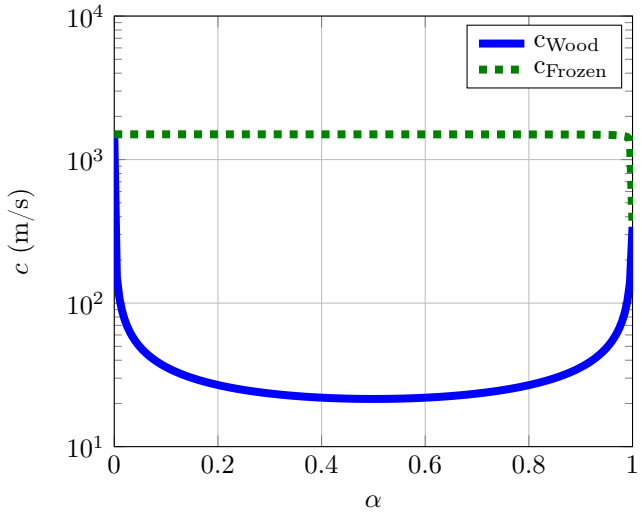


Figure 1.2: *Frozen velocity of sound (dashed line) and Wood's velocity of sound (full line) as a function of volume fraction α. The subcharacteristic condition ($c_{Frozen} \geqslant c_{Wood}$) is verified for all values of α.*

We can observe that using the second principle of thermodynamics has lead to a 4-equation system, where the phase pressures can now be out of the mechanical equilibrium, since α is now an independent variable that may not satisfy the pressure equality $p_1\left(\frac{\rho Y}{\alpha}\right) = p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right)$ any longer. However, the source term of equation (1.36d) has a relaxation effect on the pressures, with a relaxation characteristic time proportional to ϵ. This equation is similar to the equation on volume fraction (1.11) of Baer and Nunziato (1986). In their paper, the expression given to the relaxation parameter was based on an analogy with the kinetic theory for gases, but their study of the compaction effects lead to the conclusion that further study was necessary to better identify this parameter. We will propose an identification of it in the case of bubbly flows in Chapter 2.

Let us conclude this section by studying the mathematical properties of this new two-phase model.

Mathematical properties:

[P4] System (1.36) is closed: ρ, Y, $\mathbf{u}$ and α have their own evolution equations, while p_1 and p_2 are computed using their pressure laws.

[P5] In one dimension, the convective part of system (1.36) (*i.e.* the left hand side of the equations) is hyperbolic with three characteristic velocities : u (eigenvalue of multiplicity 2), $u - \mathrm{c}_{\mathrm{Frozen}}$ and $u + \mathrm{c}_{\mathrm{Frozen}}$. $\mathrm{c}_{\mathrm{Frozen}}$ is named *frozen* sound velocity as it characterizes two-phase systems constantly out of pressure equilibrium. Its expression is given by:

$$\mathrm{c}_{\mathrm{Frozen}} = \left(Y c_1^2 + (1-Y)c_2^2\right)^{1/2}. \tag{1.37}$$

This acoustic velocity satisfies Whitham's subcharacteristic condition (Whitham (1974)) regarding Wood's sound velocity for the 3-equation equilibrium system (see an illustration on Figure 1.2). This sucharacteristic property is important when considering relaxation source terms. It ensures indeed that the system obtained from (1.36) for small values of ϵ is a diffusive approximation of the equilibrium system (1.30): this means that a diffusive term in $O(\epsilon)$ is added to system (1.30). This property can be proved by using a Chapman-Enskog expansion of the variables of system (1.36) as is performed in Chanteperdrix, Villedieu, and Vila (2002). We will use this property in Chapter 4 to derive relaxation numerical schemes.

Let us mention that the 4-equation system was used in Chanteperdrix, Villedieu, and Vila (2002) for numerical purposes: the Riemann problem is in fact easier to solve exactly for system (1.36) than for

system (1.30). Using the subcharacteristic property [P5], a relaxation numerical scheme can be designed (see more detail on these schemes in Chapter 4): it consists in computing the numerical fluxes for (1.36), in updating all the variables, and then in imposing the equilibrium value for the volume fraction α. This procedure corresponds to an infinitely fast relaxation, *i.e.* $\epsilon \to 0$. Therefore, there was no need for a physical meaning or for any value for ϵ.

1.3.2.4 Conclusion on the method

This last method, based on a variational principle and the second principle of thermodynamics enables to select the energies of the system, derive the associated conservative equations and add dissipative terms that comply with an entropy inequality. A first advantage of this method compared to the methods of sections 1.1 and 1.2 lies in the a priori selection of the physical effects that have to be modeled. This gives then simpler systems of equations. Moreover, the equations that are derived have interesting mathematical properties. In particular, they are closed, hyperbolic and own an entropy function. However, as for the postulated equations of section 1.2, new parameters are introduced and their physical interpretation is not obvious.

We have illustrated here the method based on the variational principle to derive two homogeneous two-fluid models, that are commonly used in the interface capturing methods. In the next section, we apply the methodology and introduce more sub-scale physics in these models.

1.4 Derivation of a hierarchy of homogeneous two-phase models

Now that we have presented the methodology (in section 1.3), let us use it to enrich the mixture models with sub-scale physical effects. These effects are naturally present in the equations issued from the averaging strategy, but in the context of the variational principle, one must associate them with energy terms.

By taking inspiration of Gavrilyuk and Saurel (2002), we are interested here in modeling the sub-scale pulsations of the interface due to pressure inequalities. This is indeed a leading physical phenomenon for bubbly flows, where the bubbles pulsations are strong sources of energy damping. We will then consider new energy terms in the Lagrangian functional, that are associated with these pulsation effects and with some other effects that will be discussed later.

In this part, although the new energy terms are inspired by the bubbly media, the resulting equations have general forms that may apply to different two-phase configurations. This is verified for the case of bubbles in Chapter 2, and propositions for more general interfacial configurations are given in Chapter 3.

1.4.1 A 5-equation system with micro-inertia and micro-viscosity

Ground hypotheses

From Section 1.3, we consider that velocity equilibrium (H1), barotropic system (H3) and no phase transition (H4) are still valid assumptions. Moreover, in mixture zones, we assume that:

(H5) the interface is submitted to *sub-scale interface pulsations*, locally modifying the volume fraction of the two phases.

Energies

The sub-scale pulsations may be considered as contributing to the kinetic energies (because of the motion of the interface) or as an internal energy from a macroscopic point of view. We choose the first interpretation and thus assume that the two-phase system owns a two-scale kinetic energy, composed by

the bulk kinetic energy $\frac{1}{2}\rho|\mathbf{u}|^2$ and this pulsational energy that we write:

$$E_{\text{puls}} = \frac{1}{2}\nu|D_t\alpha|^2. \tag{1.38}$$

ν is the inertia associated with the motion of the interface, and is called *micro-inertia*. We assume in a first approximation that ν depends on α only, by analogy with Temkin (2005), where the inertia associated with the pulsation of one bubble is a function of its radius only. And in extension, we assume it to be true for any two-phase configuration. We will see more precisely the assumptions that are made and the expression of this term in Chapter 2 for the case of bubbles.

The potential energies are now composed of the Helmholtz free energy $\rho f(\rho, Y, \alpha)$ and of a *configuration* energy $\alpha \mapsto e(\alpha)$, as referenced in Gavrilyuk and Saurel (2002). This last energy corresponds to the configuration pressure that appears in the equation for volume fraction in the BN model (1.11). We recall that the BN model applies to granular flows. In the case of gas-liquid flows, we may use this energy to describe the surface tension effects. These surface effects are related to the form of the interface. So, in a first approach, we can assume that this energy only depends on α (it is our unique information about the interface). To conclude, the Lagrangian writes:

$$\mathcal{L}(\mathbf{u}, \rho, Y, \alpha, D_t\alpha) = \frac{1}{2}\rho|\mathbf{u}|^2 + \frac{1}{2}\nu|D_t\alpha|^2 - \rho f(\rho, y, \alpha) - \rho e(\alpha). \tag{1.39}$$

Least Action Principle and Conservative Equations

We follow the same procedure as in 1.3. First, the infinitesimal variations of ρ, Y and $\mathbf{u}$ are given by the relations (1.22). One can show that the variations of $D_t\alpha$ are related to the variations of α by the formula:

$$\delta(D_t\alpha) = D_t(\delta\alpha) + (\boldsymbol{\nabla}\alpha)^T\left[D_t\boldsymbol{\xi} - \left(\boldsymbol{\xi}^T\boldsymbol{\nabla}\right)\mathbf{u}\right].$$

The infinitesimal variations of the Hamiltonian action are then given by:

$$\begin{aligned}\delta\mathcal{A} = -\int_\Omega \Bigg\{ &-\rho\boldsymbol{\nabla}\left(\frac{\partial\mathcal{L}}{\partial\rho}\right) + \partial_t\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right) + \operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\mathbf{u}^T\right) + (\boldsymbol{\nabla}\mathbf{u})^T\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right) + \left(\frac{\partial\mathcal{L}}{\partial Y}\right)\boldsymbol{\nabla}Y \\ &+ \left[\partial_t\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\right) + \operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\mathbf{u}\right)\right]\boldsymbol{\nabla}\alpha + \frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\boldsymbol{\nabla}(D_t\alpha)\Bigg\}^T \boldsymbol{\xi}\,\mathrm{d}\mathbf{x}\mathrm{d}t \\ -\int_\Omega &\left[-\frac{\partial\mathcal{L}}{\partial\alpha} + \partial_t\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\right) + \operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\mathbf{u}\right)\right]\delta\alpha\,\mathrm{d}\mathbf{x}\mathrm{d}t.\end{aligned} \tag{1.40}$$

and according to the Least Action Principle, we have the following relations that define the extremum of the action:

$$\begin{aligned}-\rho\boldsymbol{\nabla}\left(\frac{\partial\mathcal{L}}{\partial\rho}\right) + \partial_t\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right) + \operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\mathbf{u}^T\right) + (\boldsymbol{\nabla}\mathbf{u})^T\left(\frac{\partial\mathcal{L}}{\partial\mathbf{u}}\right) + \left(\frac{\partial\mathcal{L}}{\partial Y}\right)\boldsymbol{\nabla}Y& \\ + \left[\partial_t\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\right) + \operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\mathbf{u}\right)\right]\boldsymbol{\nabla}\alpha + \frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\boldsymbol{\nabla}(D_t\alpha) &= \mathbf{0}, \quad (1.41) \\ -\frac{\partial\mathcal{L}}{\partial\alpha} + \partial_t\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\right) + \operatorname{div}\left(\frac{\partial\mathcal{L}}{\partial(D_t\alpha)}\mathbf{u}\right) &= 0. \quad (1.42)\end{aligned}$$

Finally relations (1.41) and (1.42) turn out to be, by using the definition of the Lagrangian (1.39):

$$\begin{aligned}&\partial_t(\rho\mathbf{u}) + \operatorname{div}(\rho\mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla}\left(\rho^2\frac{\partial f}{\partial\rho} + \frac{1}{2}\nu(D_t\alpha)^2\right) = \mathbf{0} \\ &-\frac{1}{2}(D_t\nu)(D_t\alpha) + \partial_t(\nu D_t\alpha) + \operatorname{div}(\mathbf{u}\nu D_t\alpha) = -\rho\frac{\partial f}{\partial\alpha} - \rho\frac{\partial e}{\partial\alpha}.\end{aligned}$$

Let us now introduce the mixture pressure $p = \rho^2\frac{\partial f}{\partial\rho}$ and an auxiliary variable w defined by:

$$D_t\alpha = \frac{\rho Y w}{\sqrt{\nu}}. \tag{1.43}$$

The conservative system of equations writes:

$$\partial_t \rho + \operatorname{div}(\rho \mathbf{u}) = 0, \tag{1.44a}$$
$$\partial_t (\rho Y) + \operatorname{div}(\rho Y \mathbf{u}) = 0, \tag{1.44b}$$
$$\partial_t (\rho \mathbf{u}) + \operatorname{div}(\rho \mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla}\left(p + \frac{1}{2}\nu (D_t \alpha)^2\right) = \mathbf{0}, \tag{1.44c}$$
$$\partial_t \alpha + \mathbf{u}^T \boldsymbol{\nabla}\alpha = \frac{\rho Y w}{\sqrt{\nu}}, \tag{1.44d}$$
$$\partial_t w + \mathbf{u}^T \boldsymbol{\nabla} w = -\frac{1}{Y\sqrt{\nu}}\left(\frac{\partial f}{\partial \alpha} + \frac{\partial e}{\partial \alpha}\right). \tag{1.44e}$$

Entropy dissipation

In the next step, we aim at describing sub-scale dissipation associated with the pulsation kinetic energy E_{puls} of (1.38). Indeed, in the case of bubbles, the damping mechanisms associated with bubbles pulsations are well-known (see the large study conducted in Prosperetti (1977) and the beginning of Chapter 2). In a general two-phase flow context, we propose to introduce irreversible damping effects too, by adding terms to (1.44) that are compatible with a mathematical entropy evolution equation.

As already seen in section 1.3.2.3, an entropy function for isothermal systems is given by the total energy. Thus, for system (1.44), it writes:

$$\rho\eta(\mathbf{u}, \rho, Y, \alpha, w) = \frac{1}{2}\rho|\mathbf{u}|^2 + \frac{1}{2}(\rho Y w)^2 + \rho f(\rho, Y, \alpha) + \rho e(\alpha). \tag{1.45}$$

In the same way as in section 1.3.2.3, we assume that equations (1.44a)-(1.44d) are valid, and discard (1.44e): $D_t w$ is now a quantity to be defined.

While looking for an entropy flux function $\mathbf{G}$ that satisfies

$$\partial_t(\rho\eta) + \operatorname{div}(\rho\eta\mathbf{u} + \mathbf{G}) \leqslant 0,$$

the development of the entropy equation leads to:

$$\operatorname{div}(\mathbf{G}) + \rho\left(\frac{\partial \eta}{\partial \mathbf{u}}\right)^T D_t \mathbf{u} + \rho\frac{\partial \eta}{\partial \rho} D_t \rho + \rho\frac{\partial \eta}{\partial Y} D_t Y + \rho\frac{\partial \eta}{\partial \alpha} D_t \alpha + \rho\frac{\partial \eta}{\partial w} D_t w \leqslant 0.$$

Then, using (1.44a)-(1.44d) to express $D_t\mathbf{u}$, $D_t\rho$, D_tY, $D_t\alpha$ and

$$\frac{\partial \eta}{\partial \mathbf{u}} = \mathbf{u}, \qquad \frac{\partial \eta}{\partial \alpha} = \frac{\partial f}{\partial \alpha} + \frac{\partial e}{\partial \alpha}, \qquad \frac{\partial \eta}{\partial \rho} = \frac{1}{2}(Yw)^2 + \frac{\partial f}{\partial \rho}, \qquad \frac{\partial \eta}{\partial w} = \rho Y^2 w,$$

we finally find that

$$\operatorname{div}\left[\mathbf{G} - \left(p + \frac{1}{2}(\rho Y w)^2\right)\mathbf{u}\right] + \frac{\rho^2 Y w}{\sqrt{\nu}}\left(\frac{\partial f}{\partial \alpha} + \frac{\partial e}{\partial \alpha} + \sqrt{\nu} Y D_t w\right) \leqslant 0. \tag{1.46}$$

Once more, we choose to ensure (1.46) by setting

$$\mathbf{G} = \left(p + \frac{1}{2}(\rho Y w)^2\right)\mathbf{u} \quad \text{and} \quad \frac{\partial f}{\partial \alpha} + \frac{\partial e}{\partial \alpha} + \sqrt{\nu} Y D_t w = -\epsilon \frac{Y w}{\sqrt{\nu}},$$

where $\epsilon > 0$ is a constant parameter. This yields a definition of $\mathbf{G}$ and a new evolution equation for w that reads:

$$\partial_t w + \mathbf{u}^T \boldsymbol{\nabla} w = -\frac{\epsilon}{\nu} w - \frac{1}{\sqrt{\nu} Y}\left(\frac{\partial f}{\partial \alpha} + \frac{\partial e}{\partial \alpha}\right).$$

Consequently, the generic form of our two-phase flow system reads:

$$\partial_t \rho + \mathrm{div}(\rho \mathbf{u}) = 0, \tag{1.47a}$$
$$\partial_t (\rho Y) + \mathrm{div}(\rho Y \mathbf{u}) = 0, \tag{1.47b}$$
$$\partial_t (\rho \mathbf{u}) + \mathrm{div}(\rho \mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla}\left(p + \frac{1}{2}(\rho Y w)^2\right) = \mathbf{0}, \tag{1.47c}$$
$$\partial_t \alpha + \mathbf{u}^T \boldsymbol{\nabla}\alpha = \frac{\rho Y w}{\sqrt{\nu}}, \tag{1.47d}$$
$$\partial_t w + \mathbf{u}^T \boldsymbol{\nabla} w = -\frac{\epsilon}{\nu} w - \frac{1}{Y\sqrt{\nu}}\left(\frac{\partial f}{\partial \alpha} + \frac{\partial e}{\partial \alpha}\right). \tag{1.47e}$$

Final form of the system

For the choice of the free energy given in (1.29), this system finally writes:

$$\partial_t \rho + \mathrm{div}(\rho \mathbf{u}) = 0, \tag{1.48a}$$
$$\partial_t (\rho Y) + \mathrm{div}(\rho Y \mathbf{u}) = 0, \tag{1.48b}$$
$$\partial_t (\rho \mathbf{u}) + \mathrm{div}(\rho \mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla}\left(p + \frac{1}{2}\nu(D_t\alpha)^2\right) = \mathbf{0}, \tag{1.48c}$$
$$\partial_t \alpha + \mathbf{u}^T \boldsymbol{\nabla}\alpha = \frac{\rho Y w}{\sqrt{\nu}}, \tag{1.48d}$$
$$\partial_t w + \mathbf{u}^T \boldsymbol{\nabla} w = -\frac{\epsilon}{\nu} w + \frac{1}{\rho Y\sqrt{\nu}}\left(p_1 - p_2 - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha}\right). \tag{1.48e}$$

The new variable w accounts for volume fraction variations due to small scale velocities. It is worth noting these vibration-like effects impact the total momentum of the mixture and appear as an additional pressure term in the third equation of system (1.47).

Let us now present some mathematical properties for system (1.48).

Mathematical properties

[P6] System (1.48) is closed, assuming ϵ and μ are known.

[P7] In one dimension, the convective part of system (1.48) is hyperbolic with three characteristic velocities: u (eigenvalue of multiplicity 3), $u - \mathrm{c_{Puls}}$ and $u + \mathrm{c_{Puls}}$, where c is the acoustic velocity given by:

$$\mathrm{c_{Puls}} = \left(\mathrm{c}^2_{\mathrm{Frozen}} + \rho(Yw)^2\right)^{1/2} \tag{1.49}$$

Once again, the sub-scale vibrations have an influence on the macroscopic structure of the system, here, by appearing in the acoustic velocity.

From now on, we start with model (1.48) and we examine two limit flow regimes, obtained for vanishing values of the parameters ϵ and ν. First, we study the case of a negligible micro-inertia compared to the internal dissipation effects, *i.e.* $\nu \to 0$ and $\epsilon = O(1)$. Second, we consider the case when both micro-inertia and internal dissipation tend to zero, with $\nu \to 0$ and $\epsilon = \mathrm{o}(\sqrt{\nu})$. Both cases allow to recover the two-phase system derived in section 1.3.2.2. All the systems that are examined are composed of a conservative part and a (possibly null) stiff source term.

1.4.2 A 5-equation non-dissipative system with micro-inertia

Another and simpler asymptotic limit is obtained when $\epsilon \to 0$ in (1.48). A simple calculation leads to the 5-equation conservative system given in (1.44). This system is not dissipative, but has the same

mathematical properties as system (1.48). Nonetheless, it will be studied in Chapter 2, as it isolates the effects of the micro-inertia ν from the effects of the micro-viscosity ϵ.

1.4.3 A 4-equation dissipative system

We consider system (1.48) and suppose that $\nu \to 0$ for a fixed value of ϵ. We see that (1.48e) provides that:

$$\frac{m_1 w}{\sqrt{\nu}} = \frac{1}{\epsilon}\left(p_1 - p_2 - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha}\right).$$

By using equation (1.48d) we obtain that the limit regime is governed by

$$\partial_t \rho + \mathrm{div}(\rho \mathbf{u}) = 0, \tag{1.50a}$$
$$\partial_t (\rho Y) + \mathrm{div}(\rho Y \mathbf{u}) = 0, \tag{1.50b}$$
$$\partial_t (\rho \mathbf{u}) + \mathrm{div}(\rho \mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla} p = \mathbf{0}, \tag{1.50c}$$
$$\partial_t \alpha + \mathbf{u}^T \boldsymbol{\nabla}\alpha = \frac{1}{\epsilon}\left(p_1 - p_2 - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha}\right). \tag{1.50d}$$

In the specific case $\mathrm{d}e/\mathrm{d}\alpha = 0$, we recover the relaxation system (1.36d), studied in Chanteperdrix, Villedieu, and Vila (2002). Its mathematical properties have been presented at the end of section 1.3.2.3, and we just recall here that its characteristic acoustic velocity is $\mathrm{c}_{\mathrm{Frozen}}$, and a subcharacteristic condition is satisfied between systems (1.48) and (1.50), as we have: $\mathrm{c}_{\mathrm{Frozen}} \leqslant \mathrm{c}_{\mathrm{Puls}}$.

1.4.4 A 3-equation equilibrium system

The last model of the hierarchy can be obtained either by considering model (1.48) and the two vanishing coefficients $\nu \to 0$ and $\epsilon = \mathrm{o}(\sqrt{\nu})$, or by taking $\epsilon \to 0$ in (1.50).

By multiplying each side of (1.50d) by ϵ^2 and then letting $\epsilon \to 0$, one obtains:

$$p_1\left(\frac{\rho Y}{\alpha}\right) - p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right) - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\alpha) = 0.$$

The whole system then writes:

$$\partial_t \rho + \mathrm{div}(\rho \mathbf{u}) = 0, \tag{1.51a}$$
$$\partial_t (\rho Y) + \mathrm{div}(\rho Y \mathbf{u}) = 0, \tag{1.51b}$$
$$\partial_t (\rho \mathbf{u}) + \mathrm{div}(\rho \mathbf{u}\mathbf{u}^T) + \boldsymbol{\nabla} p = \mathbf{0}, \tag{1.51c}$$
$$p_1\left(\frac{\rho Y}{\alpha}\right) - p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right) - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\alpha) = 0. \tag{1.51d}$$

and in the specific case $\mathrm{d}e/\mathrm{d}\alpha = 0$, we recover the pressure equilibrium closure relation $p_1 = p_2$ from model (1.30) and the 3-equation system which was studied in Chanteperdrix, Villedieu, and Vila (2002).

1.4.5 Conclusion on the hierarchy of two-phase models

With system (1.51), we have reached the model describing the gas-liquid mixture at its equilibrium state: pressures instantly balance out and there is no dissipation. The different asymptotic limits are summarized and illustrated in Figure 1.3. In this figure,the systems are written in a form that makes appear the source terms, parametrized by ϵ and ν.

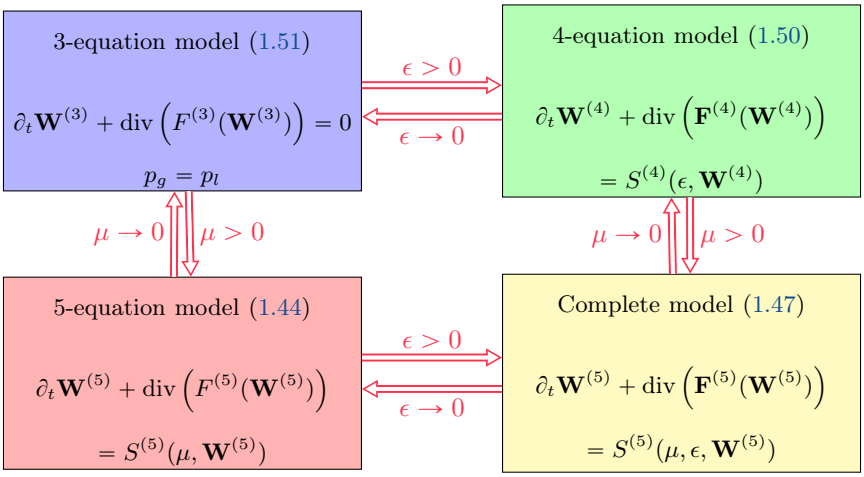


Figure 1.3: *Schematic representation of the hierarchy of models. Submodels are linked by different parameters limits.*

1.5 Conclusion and perspectives

In this chapter, we have compared three methods that are used in the literature to derive two-fluid and mixture models:

- the first one is based on the averaging of the continuum equations and the jump relations through the interface at a scale that is not resolved (the sub-scale). After being closed, the equations that are thus derived describe a large number of physical mechanisms, whose effective effect is hard to determine for a given flow configuration,
- the second method uses postulated forms of the mixture equations and performs the closure of the different dissipative mechanisms under the constraints imposed by the second law of thermodynamics. This involves the definition of new parameters, whose values can be used to derive sub-models, but whose physical meaning is not known a priori,
- the third method involves a variational principle to get the conservative parts of the equations. Dissipative structures are then added so as to satisfy the second law of thermodynamics. As for the second method, the meaning of the parameters that are introduced has to be determined a posteriori.

We have used the third method to derive a hierarchy of homogeneous two-fluid models. The most complex of these models takes into account micro-vibrations and damping of these vibrations. By letting the parameters tend to zero, one recovers successively models with dissipation through pressure relaxation and a model that is no more dissipative. This last model corresponds to the equations that are usually used in interface capturing methods. We have then drawn a path from this 3-equation model to an enriched 5-equation model where a first series of sub-scale effects are accounted for.

The procedure using the variational principle seems promising for a further enrichment of the mixture models, or, as performed in Gavrilyuk and Saurel (2002), for considering two-fluid models with two velocities. Now, we must clarify a last point, that is the physical meaning of the micro-inertia ν and ϵ parameters. This is the objective of Chapter 2.

Chapter 2

Modeling of pulsating bubbly flows

We recall that in Chapter 1, our approach consisted in enriching a two-phase mixture model, usually used for interfacial or separated-phase flows, by adding some sub-scale mechanisms, so as to better model the unresolved interactions between both phases. These sub-scale mechanisms were inspired by the dynamics of pulsating bubbles, but they were given in a general form that does not especially involve the topology of the bubbles. This leads to a hierarchy of models of increasing complexity, containing new energies and new dissipation terms related with new parameters: a micro-inertia ν and a micro-viscosity ϵ.

These two parameters must now be identified. To do so, one can compare the models with already existing averaged equations, or with experimental measures, and hence identify the parameters or fit their values with the experimental data. This identification necessitates to assume a certain flow configuration, and may not be relevant in any other context. However, this approach can bring interesting information about the sub-scale mechanisms and help design more general models. In this chapter, we consider the context of bubbly flows. We first study in more details the mechanisms involved in the pulsations of bubbles. Two models will be retained: the Rayleigh-Plesset's equation, as used in the model of Cheng, Drew, and Lahey (1985), and the model for small-amplitude pulsations of Prosperetti (1977). These models allow us to propose an identification for ν and two different identifications for ϵ.

Both these identifications and the relevance of our models to describe a flow of pulsating bubbles are then tested through comparisons with experimental data. These experimental data are measures of the dispersion of acoustic waves in bubbly flows and provide interesting information about damping and resonant phenomena. The same comparisons were performed in Cheng, Drew, and Lahey (1983); Cheng, Drew, and Lahey (1985) with the model of Cheng, Drew, and Lahey (1985); Drew and Passman (1999), which is dedicated to bubbly media and includes the description of many physical mechanisms (see Chapter 1, section 1.1). We can wonder in what extent the sub-scale mechanisms accounted for in our much simpler two-phase mixture models are sufficient to reproduce the experimental results.

We will close this chapter with a study of the parameters and their roles in the propagation of acoustic waves: their influence on the phase velocity and on the damping. We will also discuss the difference between the characteristic velocities of the convective parts of the models, and the sound velocities issued from the linear acoustic study.

2.1 Physical description of pulsating bubbles

2.1.1 Review of different approaches and models

The first works on bubbly media were interested in describing the dispersion of sound waves by bubbles, so as to be able to detect submarine objects or, on the contrary, to design acoustic screens using bubbles: it was indeed observed that air bubbles were very efficient in absorbing sound waves. Consequently, these studies (Spitzer (1943); Foldy (1945) and references therein) were concerned with the interaction of bubbles with an external variation of the pressure field. With the assumption that the bubbly media

is sufficiently disperse, the behavior of a single bubble was first considered. Then, for an acoustic wave that interacts with a bubble, these studies provided the parts of the incident wave that is absorbed and the part that is scattered by the bubble, according to the acoustic wave frequency. Already at that time, three main damping contributions were identified: the acoustic, thermal and viscous damping effects. The behavior of an ensemble of bubbles, even of different sizes, was then discussed and the collective effects were obtained by integrating individual contributions and approximating the ranges of bubbles radii whose effects prevail. It was also noted that, due to the presence of the bubbles, the compressibility of the liquid phase is changed, and thus the phase velocity of the acoustic wave is also different: from very little volume fractions of gas, this phase velocity is largely reduced (Wood (1930)). These first models had been compared with experimental measures some years later, following the work of Silberman (1957): they showed a good agreement with these measures.

The modeling of the dynamics of one bubble has been further developed in the works of Plesset and Prosperetti (see a review in Plesset and Prosperetti (1977)), and a more precise description of the bubble thermodynamics, interior state, heat and mass exchanges with the liquid, has lead, among others, to improve Rayleigh's equation Rayleigh (1917). This new equation relates the variations of the bubble volume to the difference of pressure at the surface of the bubble. It is also valid out of the linear acoustic regime, as it contains non-linear terms. Rayleigh-Plesset's equation involves damping due to the liquid viscosity. As for the other damping mechanisms described in Prosperetti (1977), they are mainly related to the bubbles interior state, and are given, in this work, in their linear limits only.

Meanwhile, with the development of models designed for numerical simulations, the need of an averaged description of bubbly flows, thus at a larger scale than the scale of one bubble, have lead to the development of averaged-type evolution equations. Works of Ishii (1975) have stated the averaging theories and procedures, while works of Drew and Passman (see Drew and Passman (1999)) enabled a fine description of the physics at the liquid-gas interface and averaged flow-field equations taking into account multiple forces, like the drag, virtual mass or Basset's forces, as well as heat exchange terms. By using the Rayleigh-Plesset's equation, which is added as a closure equation to their system, the dispersion of acoustic waves described by Drew and Passman's model are in good agreement with the experimental measures of acoustic waves propagating in monodisperse flows of bubbles (see Cheng, Drew, and Lahey (1985)).

Other approaches using averaging methods have been developed, especially for taking into account bubbles interactions with each others (see Buyevich and Shchelchkova (1978); Baudoin, Thomas, Coulouvrat, and Lhuillier (2007) for example).

Finally, the method based on a variational principle and the second principle of thermodynamics, as presented in Chapter 1, was used in Gavrilyuk, Gouin, and Perepechko (1998); Gavrilyuk and Saurel (2002); Gavrilyuk (2005) to derive two-fluid models for flows of pulsating bubbles. The micro-structural equation on volume fraction thus derived is close to the Rayleigh-Plesset's equation, but with the difference that it describes the evolution of a field variable, $\alpha(t, \mathbf{x})$, and not of a sub-scale parameter $R(t)$. In these works, the micro-inertia ν and micro-viscosity ϵ parameters were given expressions that correspond to the identification we present below, when using Rayleigh-Plesset's equation. But as we will see, these identifications lack a part of the damping mechanisms. Moreover, the models issued from the variational principle have not been compared with the models issued from the averaging procedure yet: we propose here a first comparison in the linear acoustic regime.

2.1.2 Modeling of one pulsating bubble

In this section, we are interested in two models that describe the physics of one pulsating bubble:

- the first one is Rayleigh-Plesset's equation, that relates the difference of both phases pressures at the surface of the bubble, $p_{1i} - p_{2i}$, to the radius of the bubble $R(t)$, and its first and second time derivatives, $\dot{R}(t)$ and $\ddot{R}(t)$. This equation is used to close Drew's model (we recall a version of this model in 1.1, in Chapter 1),
- the second one involves in more detail the effects of the thermodynamical changes inside the bubble when it undergoes a forced oscillation. However, this model only studies the case of small amplitude pressure variations. It can then be considered in the context of this chapter, and will be used to

identify ν and ϵ, but a deeper work will probably be necessary to provide more general expressions.

Rayleigh-Plesset's equation

An equation that governs the dynamics of the bubble radius $R(t)$ can be obtained using Bernoulli's equation, relating the liquid pressure at the interface p_{2i} and the pressure far from the bubble $p_{2\infty}$, considering that the liquid is incompressible and its velocity field is irrotational. One can then introduce the velocity potential in a spherical coordinate system centered around the bubble: $\varphi : (t, r) \mapsto \varphi(t, r)$, where r is the distance from the bubble center. Then Bernoulli's relation reads:

$$\frac{p_{2i}(t)}{\rho_2} + \frac{1}{2}\left(\boldsymbol{\nabla}_r \varphi\right)^2\Big|_{r=R} + \partial_t \varphi|_{r=R} = \frac{p_{2\infty}(t)}{\rho_2} \tag{2.1}$$

A form of the potential φ is given in Landau and Lifshitz (1966) for a bubble undergoing forced oscillations by:

$$\varphi(t, r) = -\frac{R^2(t)\dot{R}(t)\,\mathrm{e}^{ik(r-R(t))}}{r(1 - ikR(t))}, \tag{2.2}$$

with k the wavelength of the oscillations in the liquid medium, and the dot notation $\dot{}$ is used for the time derivative. Moreover, Laplace relation gives the jump relation between the interfacial pressures of gas p_{1i} and liquid p_{2i}:

$$p_{1i}(t) - p_{2i}(t) = \frac{2\sigma}{R(t)} - 2\mu_2\, \partial_r(u_2(t, r))|_{r=R}\,, \tag{2.3}$$

where σ is the surface tension, u_2 is the liquid velocity component in the axial direction and μ_2 is the liquid dynamic viscosity.

By using (2.2) and (2.3) in (2.1), one obtains an equation on the bubbles radius, as can be found in Cheng, Drew, and Lahey (1985):

$$p_{1i} - p_{2\infty} = \frac{\rho_2 R\ddot{R}}{1 - ikR} + \rho_2 \dot{R}^2 \left[\frac{2}{1 - ikR} - \frac{1}{2} - \left(\frac{kR}{1 - ikR}\right)^2\right] + 4\frac{\mu_2 \dot{R}}{R}\left[1 - \frac{(kR)^2}{2(1 - ikR)}\right] + \frac{2\sigma}{R}. \tag{2.4}$$

In the limit $kR \ll 1$ (for very small bubbles or low-frequency perturbations), one recovers the classic Rayleigh-Plesset's equation Rayleigh (1917); Plesset and Prosperetti (1977):

$$p_{1i} - p_{2\infty} = \rho_2 R\ddot{R} + \frac{3}{2}\rho_2 \dot{R}^2 + \frac{4\mu_2 \dot{R}}{R} + \frac{2\sigma}{R}. \tag{2.5}$$

The next issue consists in determining the interfacial gas pressure p_{1i}, according to the state of the bubble and in particular given the bubble radius R.

Small amplitude pulsations

In the case of small amplitude oscillations of the bubble, the gas interfacial pressure variations can be related to the bubble volume and its radius variations. It is done in Prosperetti (1977) by linearizing the equations of mass, momentum and energy for the gas inside one bubble, using the equations of state and solving these linear equations using the appropriate boundary conditions at the bubble surface. This results in considering a polytropic state law for the bubble of the form:

$$\delta p_{1i} = p_{1,eq}\left[\left(\frac{R_{eq}}{R_{eq} + \delta R}\right)^{3\kappa} - 1\right] - 4\mu_{th}\dot{R},$$

where κ is a complex polytropic exponent and μ_{th} a complex thermal damping parameter, whose expressions can be found in Prosperetti (1977). R_{eq} and $p_{1,eq}$ are respectively the radius and the gas pressure

of the bubble at its equilibrium state (non oscillating), while δR and δp_{1i} are the small amplitude radius and gas pressure variations. These variables are related by:

$$R = R_{eq} + \delta R,$$
$$p_{1i} = p_{1,eq} + \delta p_{1i}.$$

Different pulsational regimes can then be studied, depending on the ratio of different effects, that are:

- the wavelength associated with the acoustic perturbation λ_{ac},
- the characteristic thermal length scale in the gas λ_{th}, given as a function of the perturbation frequency,
- the radius of the bubble R.

In particular, when $\lambda_{th}/R \ll 1$, the interior of the bubble is little affected by the acoustic perturbation, and the bubbles behave adiabatically. On the contrary, when λ_{th} is of the order of R, or greater, then the thermodynamical state of the bubble interior is changed by the external pressure perturbations, implying thermal dissipation. For very low frequencies, the regime of pulsations of the bubbles can be considered as isothermal. In general, we consider regimes where $\lambda_{ac}/R \gg 1$, in order to avoid diffraction effects.

Now, it is possible to linearize the bubbles dynamics equation (2.4), assuming that the amplitude of oscillations of the bubbles are small: $R = R_{eq}(1 + z)$, $z \ll 1$. The resulting linear equation can be assimilated to a forced damped harmonic oscillator, with the small amplitude forcing pressure δp_2, that reads (see details of the derivation in Prosperetti (1977)):

$$\rho_2 R_{eq}^2 \ddot{z} + 2\gamma\, \rho_2 R_{eq}^2 \dot{z} + \rho_2 R_{eq}^2 \omega_0^2 z = -\delta p_2 \tag{2.6}$$

The natural frequency ω_0 is given in Prosperetti (1977) by:

$$\omega_0^2 = 3\kappa p_{1,eq} \frac{1 + 2\sigma R_{eq}/p_{eq}}{\rho_2 R_{eq}^2} - \frac{2\sigma}{\rho_2 R_{eq}^3}$$

The damping coefficient γ may be split into three main contributions: $\gamma = \gamma_{\text{vis}} + \gamma_{\text{th}} + \gamma_{\text{ac}}$ (other mechanisms are considered in Cheng, Drew, and Lahey (1983)). The following expressions are found in Prosperetti (1977) for these three damping factors:

- the coefficients γ_{vis} pertains to viscous effects due to the surrounding liquid and is defined by:

  $$\gamma_{\text{vis}} = 2\mu_2/(\rho_2\, R_{eq}^2),$$

- γ_{th} is related to thermal exchanges between the gas and the liquid and writes in Prosperetti (1977):

  $$\gamma_{\text{th}} = 2\mu_{\text{th}}/(\rho_2\, R_{eq}^2),$$

  with μ_{th} a complex function depending on the bubble characteristics and the frequency of oscillations ω.

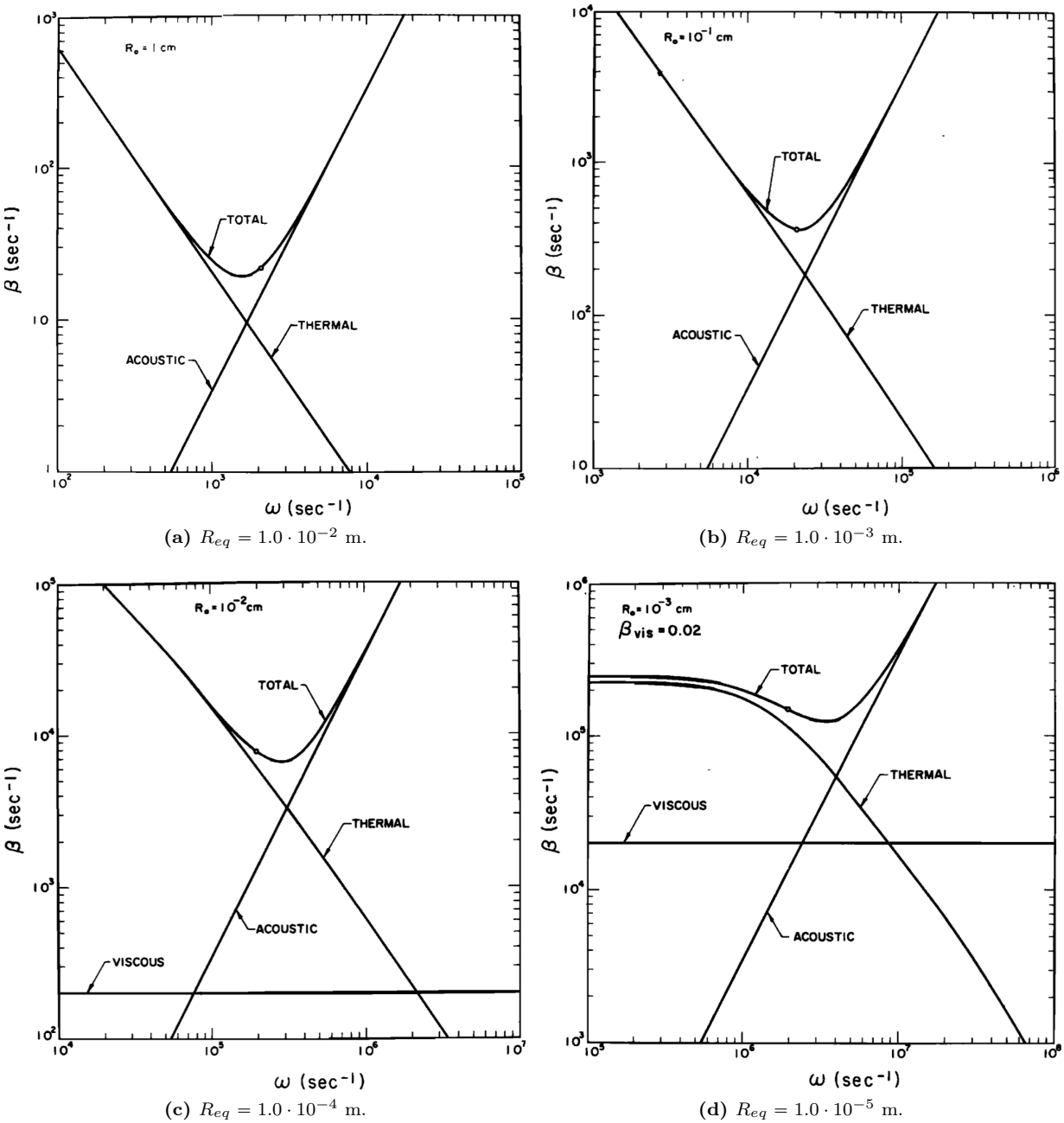
- γ_{ac} concerns acoustic scattering by the bubble and reads:

  $$\gamma_{ac} = \frac{1}{2}\omega \frac{\omega R_{eq}}{c_2} \left(1 + \frac{\omega R_{eq}}{c_2}\right)^{-1}$$

  with c_2 the velocity of sound in the liquid phase.

A review of other expressions for γ_{th} and γ_{ac} is available in Cheng, Drew, and Lahey (1983).

These three damping effects mainly depend on the sizes of the bubble, and for γ_{th} and γ_{ac} also on the frequency of the pressure perturbation. In Prosperetti (1977), one can find a study of these effects according to ω and R. This is illustrated in Figure 2.1, issued from this paper. We can then see that, for most configurations, thermal damping is the main damping effect, while viscous damping increases for smaller bubbles.



(a) $R_{eq} = 1.0 \cdot 10^{-2}$ m.

(b) $R_{eq} = 1.0 \cdot 10^{-3}$ m.

(c) $R_{eq} = 1.0 \cdot 10^{-4}$ m.

(d) $R_{eq} = 1.0 \cdot 10^{-5}$ m.

Figure 2.1: *Damping effects as functions of the pressure perturbation frequency for different bubbles sizes. The point stand at the natural frequency value. Figures from Prosperetti (1977).*

2.2 Identification of the parameters of the two-phase models in the case monodisperse bubbly flows

Let us now propose two identifications of the **micro-inertia** ν and the **micro-viscosity** ϵ parameters which were introduced in Chapter 1, by using the two models for one bubble pulsation presented in the previous section. The difficulty here lies in the fact that the two-phase systems derived in Chapter 1 use variables that characterize the flow at the effects of multiple of bubbles at the macroscopic scale. On the contrary, Rayleigh-Plesset's equation and the model from Prosperetti (1977) focus on the evolution of one bubble. We will then make some strong assumptions, namely on the sub-scale topology, that will permit us to compare the 5-equation model from Chapter 1 with the two *one-bubble* models.

2.2.1 Assumptions for a flow of bubbles

To recover the same configurations as for the models presented in the previous section, let us consider a bubbly flow which is **monodisperse** in size and characterized as follows:

- at each instant and position $(t, \mathbf{x})$, the distribution of the number of bubbles is defined by the density function $n(t, \mathbf{x})$ and all bubbles are spherical with a radius $\mathcal{R}(t, \mathbf{x})$,
- given the phasic gas density $\rho_1(t, \mathbf{x})$, all the bubbles have the same mass:

$$\mathcal{M}_b(t, \mathbf{x}) = \frac{4}{3}\pi\rho_1(t, \mathbf{x})\mathcal{R}^3(t, \mathbf{x}).$$

The gas volume fraction α and partial mass m_1 can now be related to the flow structure parameters by

$$\alpha(t, \mathbf{x}) = \frac{4\pi}{3}\mathcal{R}^3(t, \mathbf{x})n(t, \mathbf{x}), \quad m_1(t, \mathbf{x}) = n(t, \mathbf{x})\mathcal{M}_b(t, \mathbf{x}). \tag{2.7}$$

In the following, we will use the notation of the material derivative $D_t\cdot = \partial_t \cdot + \mathbf{u}^T \mathrm{div}_{\mathbf{x}}\cdot$.

Now, we make three additional assumptions:

(H1) the mass of each bubble remains constant during a medium transformation (no phase change, nor break-up, nor collapse),

(H2) the surrounding liquid is incompressible: it has a constant density $\rho_2 = \overline{\rho}_2$.

(H3) for a bubble located at $(t, \mathbf{x})$, pressure is uniform within the bubble and equal to $p_1(t, \mathbf{x})$. The pressure of the liquid surrounding the bubble is equal to $p_2(t, \mathbf{x})$.

Hypothesis **(H1)** boils down to $D_t\mathcal{M}_b = 0$. Using the conservation of the partial mass $m_1 = \rho Y$ for the gas (see Chapter 1, equation (1.48b)) and the relation (2.7), we obtain the following conservation law for n:

$$\partial_t n + \mathrm{div}_{\mathbf{x}}(n\mathbf{u}) = 0. \tag{2.8}$$

The conservation of the partial mass $m_2 = \rho(1 - Y)$ for the surrounding fluid and hypothesis **(H2)** implies that $\mathrm{div}_{\mathbf{x}}(\mathbf{u})$ is constrained by $D_t\alpha$ through the relation:

$$D_t\alpha + (\alpha - 1)\mathrm{div}_{\mathbf{x}}(\mathbf{u}) = 0. \tag{2.9}$$

2.2.2 Identification of the parameters with Rayleigh-Plesset equation

From the 5-equation system of Chapter 1, let us now express $D_t\alpha$, w and $D_t w$ in terms of $\mathcal{R}$ and its material derivatives.

By using

$$\alpha = \frac{4\pi}{3}\frac{\mathcal{R}^3 m_1}{\mathcal{M}_b},$$

we can write:

$$D_t\alpha = \alpha\left(-\mathrm{div}_{\mathbf{x}}(\mathbf{u}) + 3\frac{D_t\mathcal{R}}{\mathcal{R}}\right). \tag{2.10}$$

Then using (2.9), we obtain

$$D_t\alpha = 3\alpha(1-\alpha)\frac{D_t\mathcal{R}}{\mathcal{R}}.$$

Expressing m_1 with (2.7) and using (1.48d) leads to

$$w = \sqrt{\nu}\frac{4\pi}{\mathcal{M}_b}(1-\alpha)\mathcal{R}^2 D_t\mathcal{R}. \tag{2.11}$$

Relations (2.11) and (2.10) provide

$$D_t w = \frac{3\alpha(1-\alpha)\sqrt{\nu}}{m_1}\left[\left(2-3\alpha+\frac{3\alpha(1-\alpha)}{2}\frac{\nu'}{\nu}\right)\left(\frac{D_t\mathcal{R}}{\mathcal{R}}\right)^2 + \frac{D_{tt}^2\mathcal{R}}{\mathcal{R}}\right]. \tag{2.12}$$

Finally, combining (2.12) and (1.48e) gives the evolution equation for $\mathcal{R}$:

$$p_1 - p_2 - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha} = \epsilon\frac{3\alpha(1-\alpha)}{\mathcal{R}}D_t\mathcal{R} + 3\alpha(1-\alpha)\nu\left(3(1-\alpha)+\frac{3}{2}\alpha(1-\alpha)\frac{\nu'(\alpha)}{\nu}\right)\left(\frac{D_t\mathcal{R}}{\mathcal{R}}\right)^2 + 3\alpha(1-\alpha)\nu\frac{D_{tt}^2\mathcal{R}}{\mathcal{R}}. \tag{2.13}$$

We see that the two-phase model (1.48), supplemented by our bubbly flow structure assumptions (H1-H3), provide an evolution equation (2.13) for a spatial distribution of bubbles radii.

Now, to perform the comparison with Rayleigh-Plesset's equation, that models the evolution of one bubble, one has also to assume that:

(H4) locally, the field variable $\mathcal{R}(t,\mathbf{x})$ corresponds to the sub-scale variable $R(t)$,

and we will identify the material derivatives of $\mathcal{R}$ with the time derivatives of R: $D_t\mathcal{R} = (\mathrm{d}R/\mathrm{d}t)(t) = \dot{R}(t)$ and $D_{tt}^2\mathcal{R} = (\mathrm{d}^2R/\mathrm{d}t^2)(t) = \ddot{R}(t)$. Equation (2.13) is now an ODE, analogous to the evolution equation of a nonlinear oscillatory system with damping and forcing terms that reads:

$$p_1 - p_2 - \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha} = \epsilon\frac{3\alpha(1-\alpha)}{R}\dot{R} + 3\alpha(1-\alpha)\nu\left(3(1-\alpha)+\frac{3}{2}\alpha(1-\alpha)\frac{\nu'(\alpha)}{\nu}\right)\left(\frac{\dot{R}}{R}\right)^2 + 3\alpha(1-\alpha)\nu\frac{\ddot{R}}{R}. \tag{2.14}$$

From this analogy, ν is connected to the inertial effects, $\ddot{R}$, and justifies its name of "micro-inertia", while ϵ is related to damping, $\dot{R}$, which justifies its name of "micro-viscosity". Moreover, under the strong hypothesis **(H4)**, one can now compare the evolution equation of $R(t)$ with the Rayleigh-Plesset's equation (2.5) as presented in the previous section.

We finally make a new assumption that will simplify the identification of ν and ϵ:

(H5) we examine a flow regime involving a bubble radius and a volume fraction small enough such that $\alpha \ll 1$.

Moreover, we neglect surface tension effects in (2.5) by setting $\sigma = 0$ and relate $p_1 - p_2 - \rho\mathrm{d}e/\mathrm{d}\alpha$ in (2.13) to $p_{1i} - p_2$ in (2.5): this means that some information that relates the mean gas pressure p_1 to the gas pressure at the bubble surface p_{1i} may appear within the configuration energy e.

Remark 8. *Surface tension effects have not been discussed yet. For future works, we consider modeling them within the configuration energy e.*

Let us now identify the terms in $\dot{R}$, $\ddot{R}$ and $\dot{R}^2$, between equations (2.14) and (2.5). We respectively obtain:

$$\epsilon\frac{3\alpha}{R} = \frac{4\mu_2}{R}, \quad \frac{3\alpha}{R}\nu = \rho_2 R \quad \text{and} \quad \frac{3\alpha}{R^2}\nu\left(\frac{1}{2}\nu^{-1}\nu'(\alpha)3\alpha + 2\right) = \frac{3}{2}\rho_2. \tag{2.15}$$

The first two relations of (2.15) allow to identify respectively ϵ and ν as

$$\epsilon_{\mathrm{RP}} = \frac{4\mu_2}{3\alpha} \quad \text{and} \quad \nu_{\mathrm{RP}} = \frac{\rho_2 R^2}{3\alpha} = \frac{\rho_2 (3\alpha)^{-1/3}}{(4\pi n)^{2/3}}. \tag{2.16}$$

The third relation of (2.15) is redundant but compatible with the definition of ν expressed in (2.16).

To conclude, we have shown that for the specific flow regime defined by the hypotheses given at the beginning of this section, the 5-equation two-phase model is able to describe a flow of locally monodisperse bubbles, through equation (2.13). Moreover, this equation corresponds to the Rayleigh-Plesset equation (2.5) when the field variable $\mathcal{R}(t,\mathbf{x})$ corresponds to the sub-scale variable $R(t)$ for one bubble. The resulting value of ν and ϵ match the results of Gavrilyuk and Saurel (2002) derived in the context of a Baer-Nunziato-type two-fluid model. Let us mention similar models for micro-inertia in the literature:

- in Temkin (2005), a pulsational energy is considered in the form: $E_{puls} = \frac{1}{2} M_{puls} \dot{R}^2$ and $M_{puls} = 4\pi\rho_2 R^3$,
- an alternative approach that allows to incorporate small scale bubble velocity is also proposed in Lhuillier, Theofanous, and Liou (2010), by accounting for pseudo-turbulent kinetic energies generated by particles pulsations $K_c = \frac{1}{2} Q(\alpha)(D_t R)^2$, where $Q(\alpha)$ is proportional to 3α in the dilute limit of the disperse phase.

One strong assumption that was made here is the uniformity of pressure inside the bubbles. According to the different studies mentioned previously, this is verified only for very small bubbles and low-frequency perturbations. In the general case, thermal effects play an important role in the bubbles dynamics. Taking into account these effects in a two-phase system of equations would probably require to change the form of the internal energies, and derive the appropriate equations using the procedure of Chapter 1. But, we will see in the next section, and justify it by the comparisons with measures, that a small modification in the identification of ν and ϵ can be sufficient, in a linear regime, to recover the missing physical effects.

2.2.3 Identification of the parameters with linearized equations describing the bubble dynamics

We further examine the bubbly flow model of the previous section by considering now the regime of small variations of the bubble radius, so as to recover the configuration of the works of Prosperetti (1977) that were presented before. Let us assume that

$$\begin{array}{rclcrcl} R(t) & = & \overline{R}(1+rz(t)), & & n(t,\mathbf{x}) & = & \overline{n}, \\ p_2(t,\mathbf{x}) & = & \overline{p_2} + r\delta p_2(t), & & \mathcal{M}_b(t,\mathbf{x}) & = & \overline{\mathcal{M}_b}, \end{array} \tag{2.17}$$

where $\overline{R}$, $\overline{p_2}$, $\overline{\mathcal{M}_b}$, $\overline{n}$ are constant values and $0 < r \ll 1$ is a small parameter. If one notes:

$$\begin{array}{rclcrcl} \overline{\alpha} & = & 4\pi\overline{R}^3\overline{n}/3, & & \overline{p_1} & = & p_1(\overline{\rho_1}), \\ \overline{\rho_1} & = & \overline{n}\overline{\mathcal{M}_b}/\overline{\alpha}, & & \overline{c_1} & = & c_1(\overline{\rho_1}), \\ \overline{\rho} & = & \overline{\alpha}\,\overline{\rho_1} + (1-\overline{\alpha})\overline{\rho_2}, & & \nu(\overline{\alpha}) & = & \overline{\nu}, \end{array}$$

then (2.17) yield

$$\alpha = \overline{\alpha}(3+rz) + O(r^2), \qquad p_1 = \overline{p_1} - 3\overline{\rho_1}\,\overline{c_1}^2 rz + O(r^2) \tag{2.18a}$$

$$\nu = \overline{\nu} + O(r), \qquad \rho\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha}) = \overline{\rho}\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha}) + 3\overline{\alpha}z\left[\overline{\rho}\frac{\mathrm{d}^2 e}{\mathrm{d}\alpha^2}(\overline{\alpha}) - \overline{\rho_2}\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha})\right] r + O(r^2). \tag{2.18b}$$

Injecting (2.18) into (2.14), we obtain

$$\overline{p_1} - \overline{p_2} - r\delta p_2 - \overline{\rho}\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha}) - 3\overline{\alpha}z\left[\overline{\rho}\frac{\mathrm{d}^2 e}{\mathrm{d}\alpha^2}(\overline{\alpha}) - \overline{\rho_2}\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha})\right] r = 3r\left(\overline{\rho_1}\,\overline{c_1}^2 z + \overline{\alpha}\epsilon\dot{z} + \overline{\alpha}\,\overline{\nu}\ddot{z}\right) + O(r^2).$$

Identifying same order terms with respect to r yields

$$\overline{p_1} - \overline{p_2} - \overline{\rho}\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha}) = 0,$$

and

$$3\overline{\alpha}\,\overline{\nu}\ddot{z} + 3\overline{\alpha}\epsilon\dot{z} + 3\left[\overline{\rho_1}\,\overline{c_1}^2 + \overline{\alpha}\,\overline{p}\frac{\mathrm{d}^2 e}{\mathrm{d}\alpha^2}(\overline{\alpha}) - \overline{\alpha\rho_2}\frac{\mathrm{d}e}{\mathrm{d}\alpha}(\overline{\alpha})\right] z = -\delta p_2. \tag{2.19}$$

Equation (2.19) is a second order linear ODE in z that is consistent with the evolution of a linear harmonic oscillator with damping and forcing terms. Identifying the terms in (2.19) and (2.6) ($\overline{R} = R_{eq}$ and $\overline{\rho_2} = \rho_2$) yields the following definitions for ν and ϵ:

$$\epsilon_{\mathrm{Lin}} = \frac{4\mu_2}{3\overline{\alpha}} + \frac{2\overline{\rho_2}\overline{R}^2}{3\overline{\alpha}}\left(\gamma_{\mathrm{th}} + \gamma_{\mathrm{ac}}\right), \quad \nu_{\mathrm{Lin}} = \frac{\overline{\rho_2}\overline{R}^2}{3\overline{\alpha}}. \tag{2.20}$$

Thus, we see that the above analysis and the resulting relation (2.20) provide a definition for ϵ that is different from (2.16). More specifically, as $\epsilon_{\mathrm{Lin}} > \epsilon_{\mathrm{RP}}$ we can see that (2.20) yields a greater damping than (2.16). We recall that this difference is due to the simplifying hypotheses at the core of model (1.48) and our simple bubbly flow model (no thermal exchanges, no pressure fluctuations within the bubbles).

Now that we have identified the different sources of damping and their contributions in the micro-viscosity expression, formal expressions for γ_{th} and γ_{ac} must be given. However, in the literature, there only exists formulae for the small amplitude and forced oscillations regime. These formulae rely on the frequency of the perturbation ω and are complicated functions. In the sequel, we will use the following relations, that may not be the most physically accurate, but keep it simple:

- Pfriem's expression for γ_{th} that can be found in Cheng, Drew, and Lahey (1983):

$$\gamma_{\mathrm{th}}^{\mathrm{P40}}(\omega) = \omega_n \frac{3(\gamma_1 - 1)\sqrt{2a_1}}{2\sqrt{\omega}R}, \tag{2.21}$$

  where a_1 is the thermal diffusivity of the gas and γ_1 its ratio of specific heats,

- the natural frequency ω_n is given by the relation from Cheng, Drew, and Lahey (1983):

$$\omega_n^2 = \frac{3\kappa_1 p_1}{\rho_1 R^2}, \tag{2.22}$$

  with κ_1 the thermal conductivity of the gas,

- for γ_{ac} we consider the relation found in Prosperetti (1977):

$$\gamma_{\mathrm{ac}}^{\mathrm{P77}}(\omega) = 0.5\,\frac{\omega^2 R c_2}{c_2^2 + (\omega R)^2}. \tag{2.23}$$

Moreover, constant values for these damping effects can be considered, by evaluating the expressions (2.21) and (2.23) at the natural frequency (2.22) : $\gamma_{\mathrm{th}} = \gamma_{\mathrm{th}}^{\mathrm{P40}}(\omega_n)$ and $\gamma_{\mathrm{ac}} = \gamma_{\mathrm{ac}}^{\mathrm{P77}}(\omega_n)$.

2.2.4 Summary

In this section, we have proposed approaches to identify the micro-inertia ν and micro-viscosity ϵ parameters, by assuming a sub-scale interfacial topology that corresponds to a flow of bubbles of locally unique radius $\mathcal{R}$. This has first lead to establish an equation for $\mathcal{R}$, (2.13), and then, due to the identification of $\mathcal{R}$ with the sub-scale variable R, to the two following identifications:

- ν_{RP} and ϵ_{RP}, issued from the comparison of (2.13) with Rayleigh-Plesset equation, and for which:

$$\nu_{\mathrm{RP}} = \frac{\rho_2 R^2}{3\alpha}, \quad \text{and} \quad \epsilon_{\mathrm{RP}} = \frac{4\mu_2}{3\alpha},$$

- ν_{Lin} and ϵ_{Lin}, issued from the comparison of the linearization of (2.13) for small-amplitude variations with the works of Prosperetti (1977):

$$\nu_{\mathrm{Lin}} = \frac{\rho_2 \overline{R}^2}{3\overline{\alpha}}, \quad \text{and} \quad \epsilon_{\mathrm{Lin}} = \epsilon_{\mathrm{RP}} + \frac{2\rho_2 \overline{R}^2}{3\overline{\alpha}}\left(\gamma_{\mathrm{th}} + \gamma_{\mathrm{ac}}\right).$$

2.3 Comparisons of the two-phase models dispersion relations with experimental measures and a reference model

In order to verify that the identifications that were performed in the previous section are physically relevant, we are conducting here a first series of comparisons of the physics described by the two-phase models of Chapter 1, section 1.4 with the reference model of Drew and Passman (1999) (see 1.1 in Chapter 1) and its closure equation (2.4). These comparisons are made in the acoustic linear regimes of the equations. This also enables us to compare these two models with experimental measures of sound propagation in bubbly flows. Acoustic waves are small amplitude pressure perturbations, for which a linearized approximation of the equations can be used. Dispersion relations specific to each model are hence formulated. These analytical dispersion relations are easy to manipulate and permit to understand the physical effects involved in these particular flow regimes.

A deeper study of the whole system is certainly necessary to draw sounder conclusions, in particular for comparisons of the non-linear effects of the two models. But, this type of comparisons are far more complex than comparing dispersion relations. We consider thus the works presented below as a first step and a basis for further modeling.

2.3.1 Computing the dispersion relations for the two-phase mixture models

Considering smooth solutions of one-dimensional problems, systems (1.48), (1.50) and (1.51) can be expressed using the generic quasilinear form

$$\partial_t \mathbf{W} + \mathbf{A}(\mathbf{W})\partial_x \mathbf{W} = \mathbf{S}(\mathbf{W}). \tag{2.24}$$

Following standard lines Whitham (1974); E. Burman and L. Sainsaulieu (1995), we seek for a monochromatic wave solution of (2.24) by writting $\mathbf{W}$ in the form

$$\mathbf{W}(x,t) = \mathbf{W}^{(0)} + r\mathbf{W}^{(1)}(x,t) + O(r^2), \quad \mathbf{W}^{(1)}(x,t) = \widehat{\mathbf{W}}^{(1)} \exp\Big(i\omega t - ik(\omega)x\Big), \tag{2.25}$$

where ω is the angular frequency, k the wavelength and r is a small amplitude parameter. The equilibrium state $\mathbf{W}^{(0)}$ and the perturbation amplitude $\widehat{\mathbf{W}}^{(1)}$ both are constant. The fluid parameters associated to $\mathbf{W}^{(0)}$ are noted with the superscript $^{(0)}$ and for the sake of simplicity, we suppose that $\mathbf{W}^{(0)}$ is a steady equilibrium state, *i.e.* :

$$u^{(0)} = 0, \quad w^{(0)} = 0 \quad \text{and} \quad p_1^{(0)} = p_2^{(0)}.$$

Injecting (2.25) into (2.24) and identifying terms with respect to the powers of r yields

$$\mathbf{S}(\mathbf{W}^{(0)}) = 0, \quad \widehat{\mathbf{W}}^{(1)} \in \ker\Big(i\omega\mathbf{Id} - ik(\omega)\mathbf{A}(\mathbf{W}^{(0)}) - \mathbf{S}'(\mathbf{W}^{(0)})\Big). \tag{2.26}$$

Consequently ω and $k(\omega)$ are bound by the so-called dispersion relation:

$$\det\Big(i\omega\mathbf{Id} - ik(\omega)\mathbf{A}(\mathbf{W}^{(0)}) - \mathbf{S}'(\mathbf{W}^{(0)})\Big) = 0. \tag{2.27}$$

This allows to define the phase velocity $\mathrm{c}_{\mathrm{Phase}}$ and the spatial attenuation β of the acoustic wave by

$$\mathrm{c}_{\mathrm{Phase}}(\omega) = \mathfrak{Re}[\omega/k(\omega)], \quad \beta(\omega) = \mathfrak{Im}[k(\omega)]. \tag{2.28}$$

Let us now detail the results for each system of the hierarchy of models. Using

$$\mathsf{H}(\rho, Y, \alpha) = \rho\frac{Y(1-Y)c_1^2(\rho_1)c_2^2(\rho_2)}{\alpha^2(1-\alpha)^2},$$

the dispersion relation, the associated phase velocity $\mathrm{c}^{\varepsilon,\nu}_{\mathrm{Phase}}$ and the spatial attenuation $\beta^{\varepsilon,\nu}$ for the two-phase model with micro-inertia (1.48) read

$$\begin{aligned}\left(\frac{k^{\varepsilon,\nu}(\omega)}{\omega}\right)^2 &= \frac{\nu\omega^2 - i\epsilon\omega - \mathrm{c}^{-2}_{\mathrm{Wood}}\mathsf{H}(\rho^{(0)},Y^{(0)},\alpha^{(0)})}{\nu\mathrm{c}^2_{\mathrm{Frozen}}\omega^2 - i\epsilon\,\mathrm{c}^2_{\mathrm{Frozen}}\omega - \mathsf{H}(\rho^{(0)},Y^{(0)},\alpha^{(0)})},\\ \mathrm{c}^{\varepsilon,\nu}_{\mathrm{Phase}}(\omega) &= \mathfrak{Re}\left[\frac{\omega}{k^{\varepsilon,\nu}(\omega)}\right],\\ \beta^{\varepsilon,\nu}(\omega) &= \mathfrak{Im}[k^{\varepsilon,\nu}(\omega)].\end{aligned} \tag{2.29}$$

For the micro-inertia-free dissipative model (1.50) obtained at the limit $\nu \to 0$, $\epsilon = O(1)$, we get the dispersion relation, phase velocity $\mathrm{c}^{\varepsilon}_{\mathrm{Phase}}$ and attenuation β^{ε}:

$$\begin{aligned}\left(\frac{k^{\varepsilon}(\omega)}{\omega}\right)^2 &= \frac{i\epsilon\omega + \mathrm{c}^{-2}_{\mathrm{Wood}}\mathsf{H}(\rho^{(0)},Y^{(0)},\alpha^{(0)})}{i\epsilon\mathrm{c}^2_{\mathrm{Frozen}}\omega + \mathsf{H}(\rho^{(0)},Y^{(0)},\alpha^{(0)})},\\ \mathrm{c}^{\varepsilon}_{\mathrm{Phase}}(\omega) &= \mathfrak{Re}\left[\frac{\omega}{k^{\varepsilon}(\omega)}\right],\\ \beta^{\varepsilon}(\omega) &= \mathfrak{Im}[k^{\varepsilon}(\omega)].\end{aligned} \tag{2.30}$$

Finally for the full-equilibrium model (1.51) when $\nu \to 0$, $\epsilon = \mathrm{o}(\sqrt{\nu})$, the dispersion relation reads

$$\begin{aligned}\frac{k(\omega)^2}{\omega^2} &= \frac{1}{\mathrm{c}^2_{\mathrm{Wood}}},\\ \mathrm{c}_{\mathrm{Phase}}(\omega) &= \mathrm{c}_{\mathrm{Wood}},\\ \beta &= 0.\end{aligned} \tag{2.31}$$

We recall that $\mathrm{c}_{\mathrm{Wood}}$ and $\mathrm{c}_{\mathrm{Frozen}}$ are defined by (1.31) and (1.37).

Finally, let us mention a distinctive behavior of the acoustic waves for the system (1.48) equipped with both damping and micro-inertia: in the case of low internal dissipation *i.e.* small values of ϵ, the dispersion relation (2.29) has a resonance frequency, that is a function of the equilibrium state as:

$$\omega_{\mathrm{res}} = \frac{1}{\mathrm{c}_{\mathrm{Wood}}}\sqrt{\frac{\mathsf{H}(\rho^{(0)},Y^{(0)},\alpha^{(0)})}{\nu}}. \tag{2.32}$$

2.3.2 Comparison of the dispersion relations with reference data

From now on, we shall suppose that the compaction energy is zero: $e(\alpha) = 0$, and that the barotropic EOS for each pure fluid has the form:

$$p_k = p_k^0 + c_k^2\,(\rho_k - \rho_k^0), \tag{2.33}$$

where c_k, ρ_k^0 and p_k^0 are real constants chosen as follows:

$$\begin{aligned} p_1^0 &= 1.0\times10^5\,\mathrm{Pa}, & c_1 &= 340\,\mathrm{m.s^{-1}}, & \rho_1^0 &= 1.2\,\mathrm{kg.m^{-3}},\\ p_2^0 &= 1.0\times10^5\,\mathrm{Pa}, & c_2 &= 1500\,\mathrm{m.s^{-1}}, & \rho_2^0 &= 1000\,\mathrm{kg.m^{-3}}.\end{aligned}$$

The dispersion relations (2.29), (2.30) and (2.31) now allow us to study the responses of systems (1.48), (1.50) and (1.51) in the acoustic regime, under a forced pressure oscillation and compare them with experimental results obtained for bubbly flows. The data we shall use rely on two experimental works by Silberman (1957) and Leroy, Strybulevych, Page, and Scanlon (2008). In the sequel, elements related to the experimental data will be denoted with the superscript $^{\mathrm{ref}}$. Let us first briefly outline the framework of these studies.

The data of Silberman (1957) are considered as reference experimental results in the domain of acoustic wave propagation for bubbly flows. They have been used for comparisons with several models in the literature van Wijngaarden (1972); Cheng, Drew, and Lahey (1985); Commander and Prosperetti (1989);

Drew and Passman (1999). The experimental method proposed in Silberman (1957) consists in generating standing waves in steel pipes of various lengths, filled with a bubbly liquid. The sound is generated at one end of the pipe, while small hydrophones measure sound pressure at the other end. Measures are performed between two nodes or antinodes that allow to compute the phase velocity and the attenuation. The size distribution of the bubbles is estimated using photographs. The resulting measures were very accurate, except near the resonance frequency $\omega_{\text{res}}^{\text{ref}}$. Indeed, for ω close to $\omega_{\text{res}}^{\text{ref}}$, the evaluation of the phase velocity was not possible due to the severe attenuation of the acoustic waves.

In order to obtain data in this range of frequencies, we shall use the work of Leroy, Strybulevych, Page, and Scanlon (2008). This experimental work studies acoustic wave propagating within a thin hair gel sample containing air bubbles. The sound waves are produced at one end of the system by a transducer and measurements are performed by an hydrophone at the other end. The advantage of using the gel is that the distribution of bubbles radii and volume fractions are accurately known. According to Leroy, Strybulevych, Page, and Scanlon (2008), the difference in terms of acoustic behavior between water and gel is negligible regarding the wave dispersion. By using this set up, the results of Leroy, Strybulevych, Page, and Scanlon (2008) provide accurate data for both phase velocity and attenuation in the vicinity of $\omega_{\text{res}}^{\text{ref}}$.

2.3.2.0.1 Comparisons over a wide range of frequencies

We consider a set of measures from Silberman (1957) that involve a flow characterized by $R = 2.5\,\text{mm}$ and $\alpha = 5.84 \times 10^{-4}$. Using relations (2.16) and (2.20), we obtain the following values for the parameters of the 5-equation model (1.48) and the 4-equation model (1.50)

$$\epsilon_{\text{RP}} = 2.28, \qquad \nu_{\text{RP}} = 3.57, \tag{2.34}$$

$$\epsilon_{\text{Lin}} = 1.61 \times 10^{3}, \qquad \nu_{\text{Lin}} = 3.57. \tag{2.35}$$

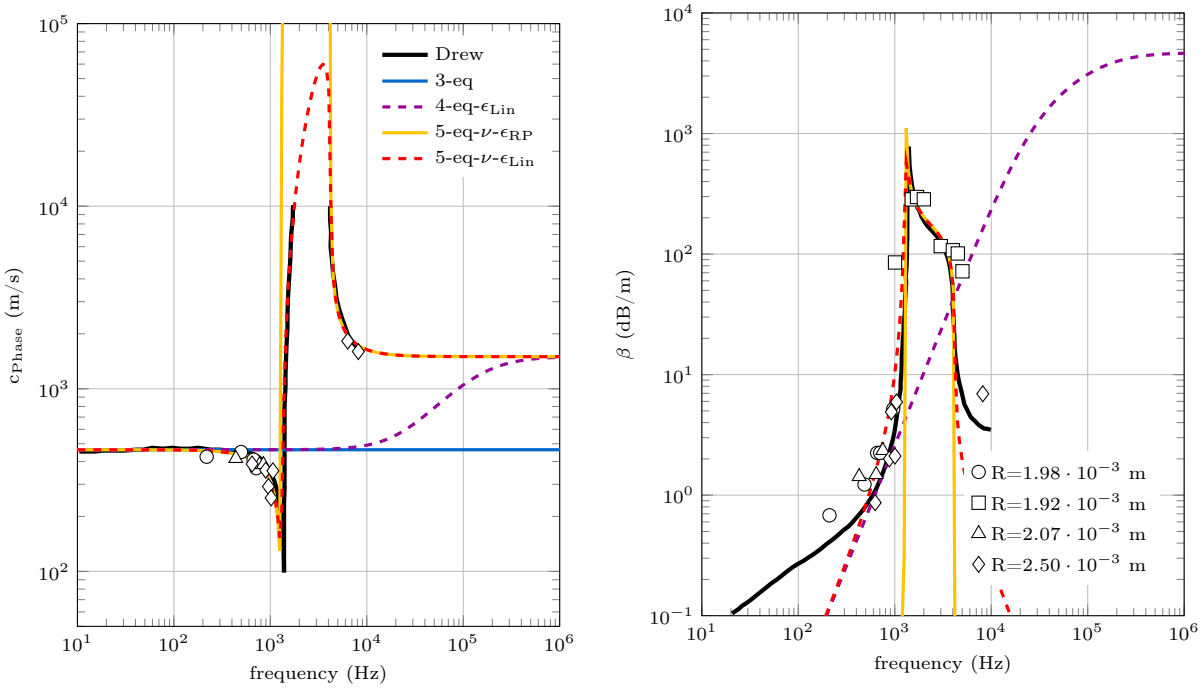
Figure 2.2 displays both phase velocity (Figure 2.2a) and spatial attenuation (Figure 2.2b), for all models of the hierarchy, for the model of Drew Cheng, Drew, and Lahey (1985) along with the experimental results.

From this comparison, one can state the following observations, for each main range of frequencies:

- **Range $\omega \ll \omega_{\text{res}}^{\text{ref}}$.** For low frequencies, the pressure perturbation is very slow and we can expect the bubbles of the system to remain at an equilibrium state with respect to both mechanics and thermodynamics. Little internal dissipation is involved in this regime, which is visible through the measures that show a low spatial attenuation. The evaluation of $c_{\text{Phase}}^{\text{ref}}$ in the experiment provides values that are close to c_{Wood}. All the models of the hierarchy show a good agreement with these data, as seen in Fig. 2.2.

  As ω increases, the spatial attenuation β^{ref} also increases. This trend is correctly followed by the models of the hierarchy that accounts for internal dissipation. Nevertheless, Prosperetti (1977) underlines that in this regime and up to a certain frequency, the thermal dissipation is the dominant internal dissipation effect (see also Figure 2.1). For all models, $\beta^{\varepsilon,\nu}$ and β^{ε} are lower than experimental data. The five-equation model (1.48) importantly underestimates $\beta^{\varepsilon,\nu}$ for $\epsilon = \epsilon_{\text{RP}}$, however for $\epsilon = \epsilon_{\text{Lin}}$ the match with the reference data is very good.

- **Range ω close to $\omega_{\text{res}}^{\text{ref}}$.** Near resonance, β^{ref} increases with ω and it becomes very large. On the contrary, $c_{\text{Phase}}^{\text{ref}}$ decreases as ω increases. It reaches $c_{\text{Phase}}^{\text{ref}} = 0$ for some frequency $\omega_{\text{ext}}^{\text{ref}} < \omega_{\text{res}}^{\text{ref}}$. For $\omega \in [\omega_{\text{ext}}^{\text{ref}}, \omega_{\text{res}}^{\text{ref}}]$, there is a good agreement between $c_{\text{Phase}}^{\varepsilon,\nu}$, $c_{\text{Phase}}^{\varepsilon}$ and $c_{\text{Phase}}^{\text{ref}}$. On the contrary, there is an important discrepancy between the phase velocity predicted by the model of Drew and $c_{\text{Phase}}^{\varepsilon,\nu}$, $c_{\text{Phase}}^{\varepsilon}$ in this range of frequencies. It is worth noting that $c_{\text{Phase}}^{\varepsilon,\nu}$ is much closer to the reference value for $\epsilon = \epsilon_{\text{Lin}}$, than for $\epsilon = \epsilon_{\text{RP}}$. Concerning the spatial attenuation, we can see that models of the hierarchy that account for micro-inertia, namely (1.48) and (1.44) fit quite well the reference results. In contrast, the models (1.50) and (1.51) yield a poor estimate of the spatial attenuation. This suggests that the source terms related to ν in (1.47d) and (1.47e) play a key role in the system behavior when $\omega \approx \omega_{\text{res}}^{\text{ref}}$. Moreover it also hints that our estimate for ν in (2.16) and (2.35) is coherent. Finally, the results suggest that the source terms related to ϵ in (1.44) do not have a great influence on the values of $\beta^{\varepsilon,\nu}$ in this range of frequencies.



(a) Phase velocity.

(b) Spatial attenuation.

Figure 2.2: *Dispersion relations for the different models of the hierarchy (full and dashed lines) and Silberman's measures (symbols) for radii of bubbles around* $R = 2.0 \cdot 10^{-3}$ *m,* $\alpha = 5.84 \cdot 10^{-4}$*,* $\epsilon_{RP} = 2.28$*,* $\epsilon_{Lin} = 1.61 \cdot 10^{3}$*,* $\nu = 3.57$*.*

- **Range** $\omega \gg \omega_{\text{res}}^{\text{ref}}$**.** For high frequencies, very few experimental data are available and thus the main comparisons element are given by the model of Drew. In this regime, one can presume that acoustic radiation effects cannot be neglected. Indeed, the bubbles starts emitting acoustic waves that are transmitted into the liquid. This process will remove energy from the bubbles and therefore will become the main damping effect of the system. For all systems of our hierarchy except (1.51), the phase velocity will tend to c_{Frozen}, which agrees with the behavior of Drew's model. The spatial attenuation coefficients of the model hierarchy do not match well the reference data of Drew's model in this range of frequencies: (1.50) clearly overestimates damping, while for (1.48) with ν_{RP} and ϵ_{RP} the dissipation is much lower. The micro-viscosity choice ϵ_{Lin} clearly increases the damping effect but it is still much lower than the dissipation of Drew's model.

In Figure 2.3 and Figure 2.4, one displays two other comparisons that confirm the good matching of the 5-equation model with experimental measures.

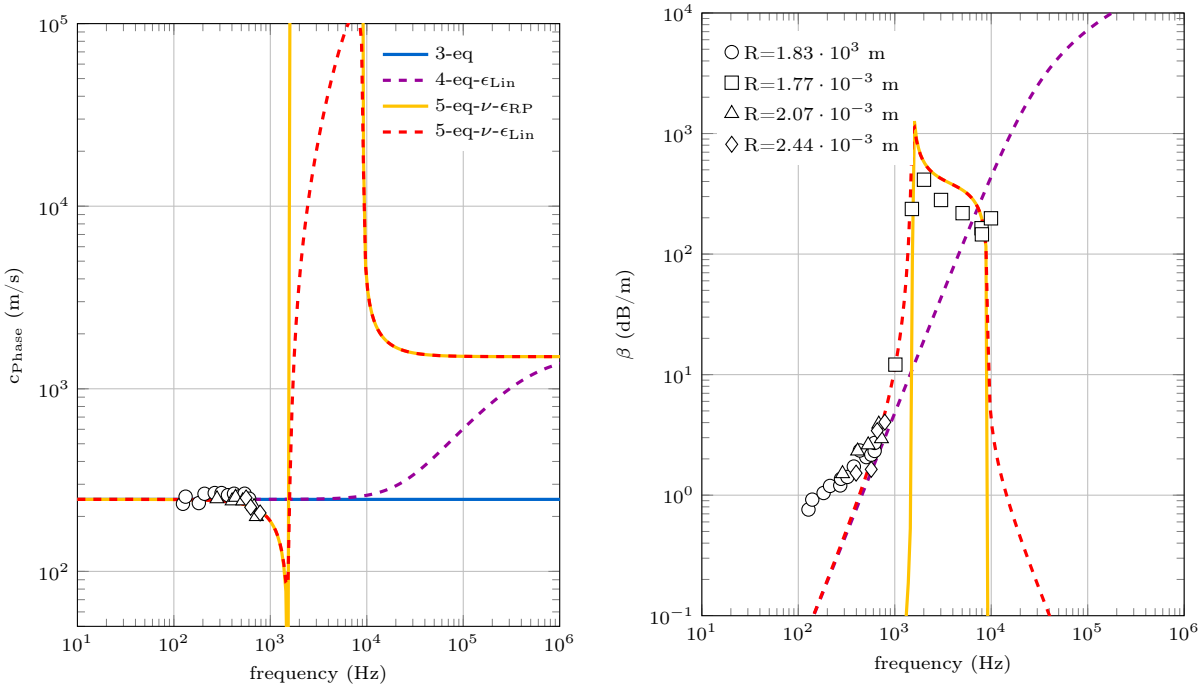
2.3.2.0.2 Finer comparison near resonance

In Leroy's experiment Leroy, Strybulevych, Page, and Scanlon (2008), the set of measures is very dense for ω close to $\omega_{\text{res}}^{\text{ref}}$. In this paragraph we discard both 4-equation model (1.50) and 3-equation model (1.51) since they cannot reproduce a resonant behavior. The bubbles in Leroy, Strybulevych, Page, and Scanlon (2008) are smaller than those of Silberman (1957), we thus consider different values of (R, α) by setting $R \approx 8.1 \times 10^{-5}$ m and $\alpha = 1.5 \times 10^{-4}$. Using (2.16) and (2.20) we obtain

$$\epsilon_{\text{RP}} = 8.89 \times 10^{-1}, \qquad \nu_{\text{RP}} = 1.46 \times 10^{-3}. \tag{2.36}$$

$$\epsilon_{\text{Lin}} = 87.3, \qquad \nu_{\text{Lin}} = 1.46 \times 10^{-3}. \tag{2.37}$$

The results we obtain with this set of parameters is coherent with the previous comparison. Indeed, in Figure 2.5 we can see that for ω close to $\omega_{\text{res}}^{\text{ref}}$, the phase velocity $c_{\text{Phase}}^{\varepsilon,\nu}$ is clearly overestimated for ϵ_{RP},

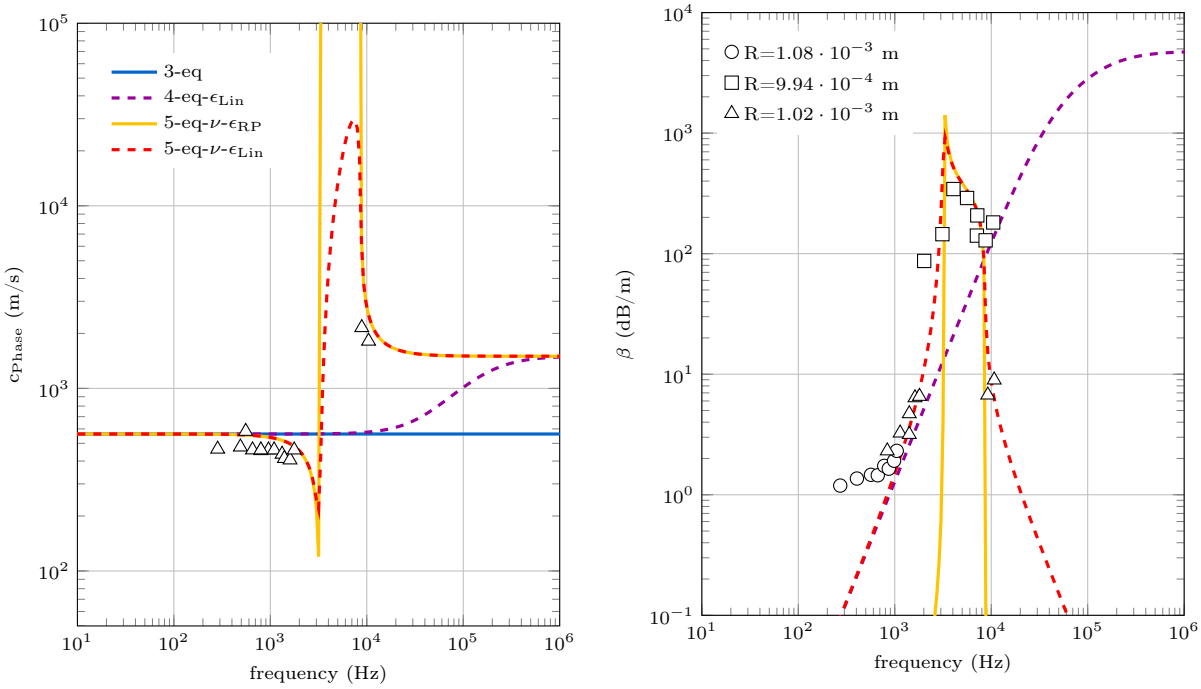


(a) Phase velocity.

(b) Spatial attenuation.

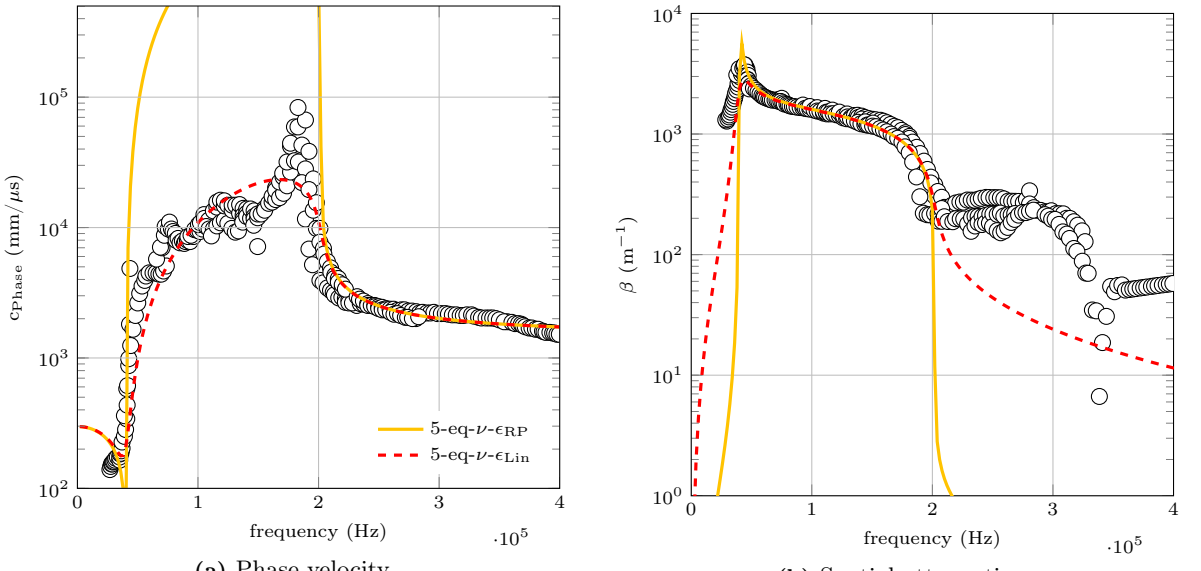
Figure 2.3: *Dispersion relations for the different models of the hierarchy (full and dashed lines) and Silberman's measures (symbols) for radii of bubbles around* $R = 2.13 \cdot 10^{-3}$ *m,* $\alpha = 2.2 \cdot 10^{-3}$*,* $\epsilon_{RP} = 6.06 \cdot 10^{-1}$*,* $\epsilon_{Lin} = 3.93 \cdot 10^{2}$*,* $\nu = 6.9 \cdot 10^{-1}$*.*

but the match with experimental data for ϵ_{Lin} seems rather accurate. Regarding the attenuation $\beta^{\varepsilon,\nu}$, the choice of ϵ_{Lin} gives clearly a better match with the reference data than ϵ_{RP}. Another comparison between the 5-equation model and experimental measures from Leroy, Strybulevych, Page, and Scanlon (2008) are given in Figure 2.6. Again, we can show a good agreement between the experimental data and the 5-equation model with the identification ϵ_{Lin}.



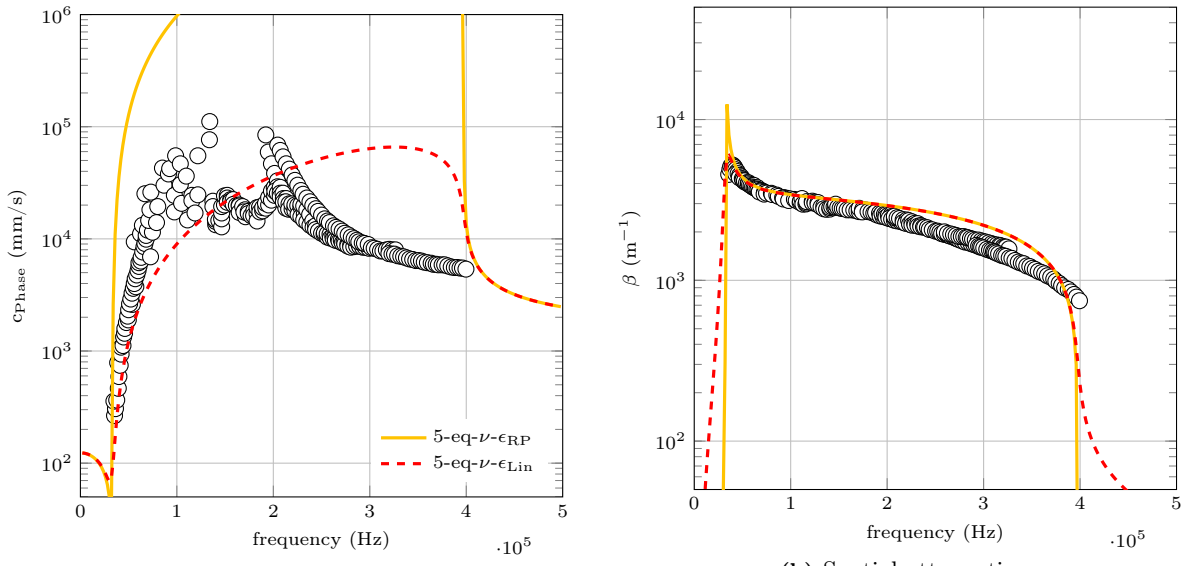
(a) Phase velocity. **(b)** Spatial attenuation.

Figure 2.4: *Dispersion relations for the different models from the hierarchy (full and dashed lines) and Silberman's measures (symbols) for radii of bubbles around* $R = 1.01 \cdot 10^{-3}$ *m,* $\alpha = 3.77 \cdot 10^{-4}$*,* $\epsilon_{RP} = 3.54$*,* $\epsilon_{Lin} = 1.58 \cdot 10^{3}$*,* $\nu = 8.95 \cdot 10^{-1}$*.*



(a) Phase velocity. **(b)** Spatial attenuation.

Figure 2.5: *Dispersion relations for the different models of the hierarchy (full and dashed lines) and Leroy's measures (symbols) for radii of bubbles around* $R = 8.1 \times 10^{-5}$ *m,* $\alpha = 1.5 \times 10^{-3}$*,* $\epsilon_{RP} = 8.89 \cdot 10^{-1}$*,* $\epsilon_{Lin} = 8.73 \cdot 10^{1}$*,* $\nu = 1.46 \cdot 10^{-3}$*.*



(a) Phase velocity.

(b) Spatial attenuation.

Figure 2.6: *Dispersion relations for the different models of the hierarchy (full and dashed lines) and Leroy's measures (symbols) for radii of bubbles around* $R = 8.1\times10^{-5}$ *m,* $\alpha = 1.5\times10^{-3}$, $\epsilon_{RP} = 1.47\cdot10^{-1}$, $\epsilon_{Lin} = 1.59 \cdot 10^{1}$, $\nu = 3.66 \cdot 10^{-4}$.

2.4 Study of the influence of the parameters

As we can note when drawing the phase velocity c_{Phase} and spatial attenuation β with respect to the frequency, the models of the hierarchy presented at the end of Chapter 1 seem to have their own range of validity in the frequency space, that corresponds to the physics they can model. In this last section, we discuss some aspects of this hierarchy of two-phase models, that we raise from their linear analysis.

Characteristic and phase velocities

Let us first note that, when accounting for internal damping with $\epsilon > 0$ and with micro-inertia (resp. without micro-inertia), the phase velocity of the acoustic wave $c^{\varepsilon,\nu}_{Phase}$ (resp. c^{ε}_{Phase}) is different from the sound velocity c_{Puls} (resp. c_{Frozen}) which is issued from the characteristic velocities of the convective parts of the systems of equations (when source terms are discarded). This underlines the fact that micro-inertia and damping source terms have a substantial influence on the phase velocity of the acoustic wave:

- for all models, the low-frequency regime is associated with c_{Wood},
- for the 4-equation and 5-equation systems, the high frequency phase velocity is c_{Frozen},
- for the 4-equation system, the transition between these two regimes is smooth,
- for the 5-equation system, there is a resonance regime, where the bubbles vibrations, driven by ν, influence the phase velocity.

We can also note that, considering the characteristic velocities, there is no asymptotic limit of c_{Frozen} towards c_{Wood} when the micro-viscosity parameter ϵ tends to zero. On the contrary, in the frequency domain, these limits are recovered. More generally, one recovers the dispersion relations from one model to its asymptotic limit model by letting the parameters ν and ϵ tend to 0.

Asymptotic limit along the parameters

Let us examine the variations of the dispersion relations across the hierarchy. As ν and ϵ reach their asymptotic limits, the transition from one model to another materializes through the dispersion rela-

tions (2.29), (2.30) and (2.31). Indeed, we see that

$$\lim_{\nu\to 0} c^{\varepsilon,\nu}_{\text{Phase}} = c^{\varepsilon}_{\text{Phase}}, \qquad \lim_{\substack{\nu\to 0\\ \epsilon=o(\sqrt{\nu})}} c^{\varepsilon,\nu}_{\text{Phase}} = c_{\text{Phase}}, \qquad \lim_{\epsilon\to 0} c^{\varepsilon}_{\text{Phase}} = c_{\text{Phase}}.$$

Le us note that these limits are not uniform over all frequencies, but they are simple limits over all bounded range of frequencies. This is illustrated in Figure 2.7. Indeed, we can observe that, whatever the value of ϵ, there is a frequency above which $c^{\varepsilon}_{\text{Phase}}$ will be close to c_{Frozen}. However, this critical frequency increases when ϵ decreases.

The transition from $c^{\varepsilon,\nu}_{\text{Phase}}$ to $c^{\varepsilon}_{\text{Phase}}$ is however uniform, as illustrated in Figure 2.8. When ν decreases, the damping effects due to ϵ prevail, and although the resonant frequency only depends on ν, the effects of resonance are completely attenuated because of ϵ. One can also note that the asymptotic behavior for spatial attenuation at low frequencies does not depend on the value of ν, but is related to the value of ϵ for the 4-equation model.

Asymptotic limit along the frequencies

The hierarchical structure of the models also shows through when one spans frequency values ω. Indeed, if one considers acoustic waves at low frequencies $\omega \ll 1$, then the dispersion relations (2.29), (2.30) and (2.31) yield

$$\left(\frac{k^{\varepsilon,\nu}(\omega)}{\omega}\right)^2 = \left(\frac{k^{\varepsilon}(\omega)}{\omega}\right)^2 + O(\omega^2) = \frac{1}{c^2_{\text{Wood}}} + O(\omega).$$

In terms of phase velocities and attenuation we obtain

$$c^{\varepsilon,\nu}_{\text{Phase}} = c^{\varepsilon}_{\text{Phase}} + O(\omega^2) = c_{\text{Wood}} + O(\omega), \qquad \beta^{\varepsilon,\nu} = \beta^{\varepsilon} + O(\omega^4) = O(\omega).$$

In the limit $\omega \to 0$ the phase velocity of the acoustic waves for all the models tends to c_{Wood} and the spatial attenuation vanishes.

Let us now turn to high frequencies $\omega \gg 1$. From (2.29), (2.30) and (2.31) we have

$$\left(\frac{k^{\varepsilon,\nu}(\omega)}{\omega}\right)^2 = \left(\frac{k^{\varepsilon}(\omega)}{\omega}\right)^2 + O\left(\frac{1}{\omega^2}\right) = \frac{1}{c^2_{\text{Frozen}}} + O\left(\frac{1}{\omega}\right). \tag{2.38}$$

Thus the acoustic waves of both systems equipped with internal damping are such that

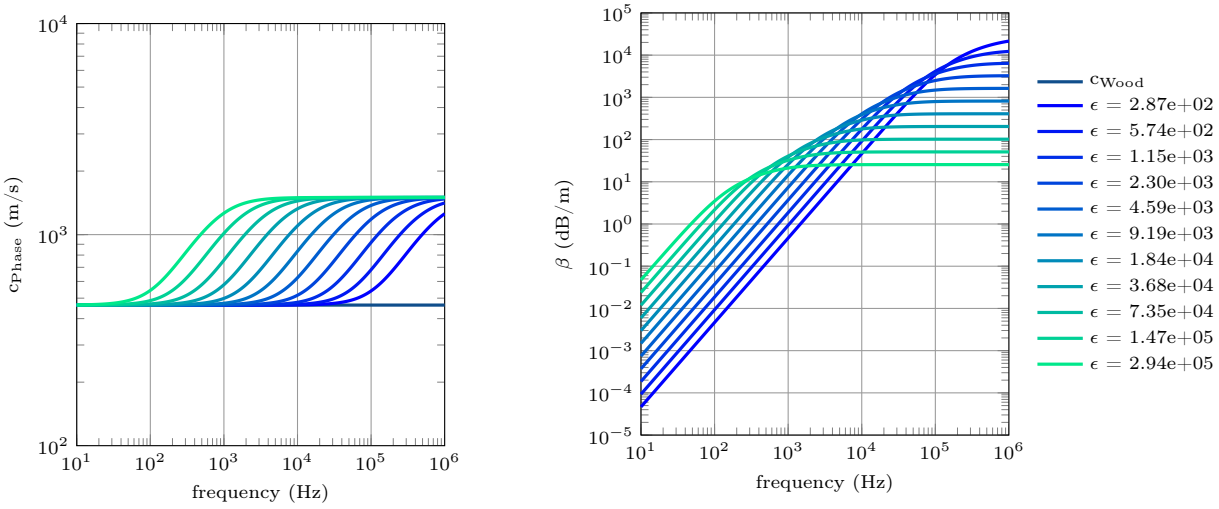
$$c^{\varepsilon,\nu}_{\text{Phase}} = c^{\varepsilon}_{\text{Phase}} + O\left(\frac{1}{\omega^2}\right) = c_{\text{Frozen}} + O\left(\frac{1}{\omega}\right). \tag{2.39}$$

Consequently, in the limit $\omega \to +\infty$, the phase velocity of the acoustic waves associated with (1.48) and (1.44) tends to c_{Frozen}, while it remains constant and equal to c_{Wood} for (1.51). In all cases the spatial attenuation tends to 0.

Conclusion on the parameters study

This study of both types of asymptotic behaviors, that are along the values of the parameters and along the frequencies, has highlighted several aspects of the hierarchical structure of the three models (1.48), (1.50) and (1.51):

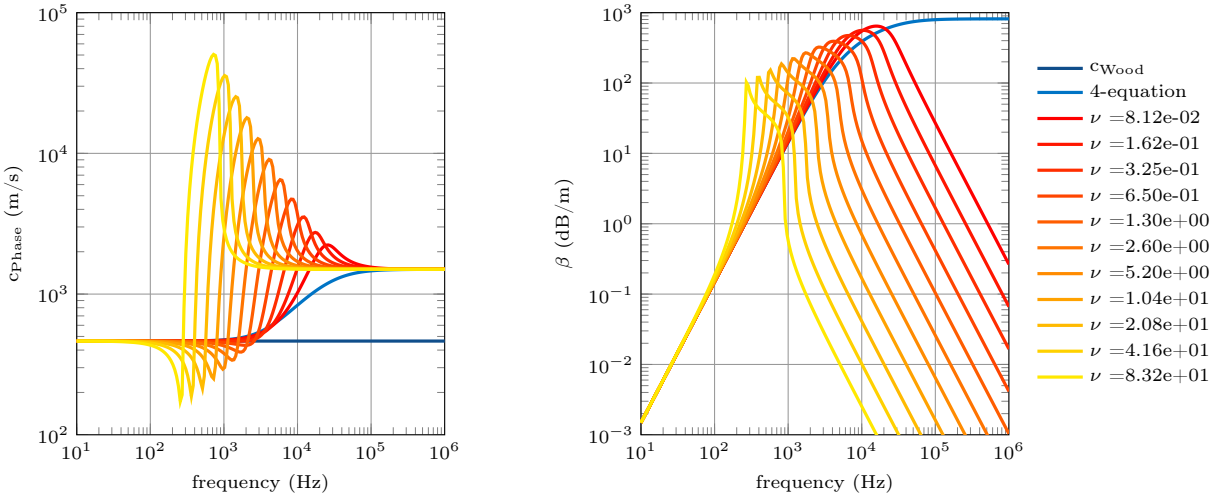
- the characteristic velocities of the convective parts of the systems do not correspond to the acoustic velocities,
- while the characteristic velocity c_{Frozen} of the 4-equation system does not tend to the characteristic velocity c_{Wood} of the 3-equation system when $\epsilon \to 0$, the phase velocity $c^{\varepsilon}_{\text{Phase}}$ tends to c_{Wood} over bounded ranges of frequencies,
- more generally, the dispersion relation for the 4-equation system does not tend uniformly towards the dispersion relation of the 3-equation system when $\epsilon \to 0$,



(a) Phase velocity.

(b) Spatial attenuation.

Figure 2.7: *Phase velocity and spatial attenuation for the 4-equation model and influence of the value of* ϵ. $\alpha = 5.8 \cdot 10^{-4}$, $R = 2.1 \cdot 10^{-3}$.



(a) Phase velocity.

(b) Spatial attenuation.

Figure 2.8: *Phase velocity and spatial attenuation for the 5-equation model and influence of the value of* ν. $\epsilon = 9.2 \cdot 10^3$, $\alpha = 5.8 \cdot 10^{-4}$, $R = 2.1 \cdot 10^{-3}$.

- this convergence is however uniform from the 5-equation dissipative system to the 4-equation system when $\nu \to 0$,
- finally, the two frequency regimes, that are the equilibrium regime at low frequencies, and the frozen regime at high frequencies, are recovered analytically for all the models (except for the 3-equation model in the frozen regime).

These properties will be used for numerical purposes (see Chapter 5).

2.5 Conclusion

First, let us recall the approach that has been chosen in Chapter 1 and in this chapter. With the objective of enriching mixture models, usually used for interface capturing methods and whose mixture zones are usually considered as numerically diffused zones, we have introduced new sub-scale mechanisms, by means of a variational principle and the second law of thermodynamics. These mechanisms stand for interface pulsations: the form of the new energy is inspired by the pulsations of bubbles and dissipation effects are associated with these pulsations. Moreover, the pulsation mechanisms are driven by two parameters: a micro-inertia ν and a micro-viscosity ϵ. An identification of these parameters has been performed through comparisons with models dedicated to the dynamics of one bubble. Finally, the identifications of ν and ϵ, as well as the relevance of the mixture models, have been assessed in the context of small-amplitude acoustic wave propagation: the dispersion relations of these models have been compared with experimental measures of sound dispersion, and with the dispersion relations for Drew's model Cheng, Drew, and Lahey (1985).

On a one hand, we can conclude from these comparisons that the 5-equation model seems to describe correctly the dispersion of sound in bubbly flows and the bubble resonance phenomenon. The pulsation energy that was postulated a priori and the identifications of the parameters are satisfactory in this context, although, we also know that some sub-scale effects, related to the thermodynamics of the bubble interior, could be better modeled. This study has permitted to determine the prevailing sub-scale mechanisms and has brought some idea for a further modeling of these mechanisms in a configuration more general than spherical monodisperse bubbles. Going further in the sub-scale modeling is the objective of Chapter 3, where the geometry of the interface, and the evolution of its deformation are studied.

On the other hand, the study of the dispersion relations has also shown the ranges of validity for the other models of the hierarchy: the 3-equation and 4-equation systems. The asymptotic limits from one model to the other can be observed in the dispersion relations. These limits appear either through vanishing values of ν and ϵ, or for low and high frequency regimes. Thus, the 3-equation system characterizes the low-frequency regime, where bubbles are constantly close to their equilibrium state. Sound propagates at c_{Wood} velocity. The 4-equation system characterizes a higher frequency regime, just below the resonance frequency. Resonance is described only by the 5-equation system, due to its inertial terms. Finally, at high frequencies, bubbles are maintained out of equilibrium, and sound propagates at c_{Frozen} velocity. We will try to recover and discuss these properties numerically in Chapter 5.

Chapter 3

Geometrical description of interfaces and connexion with spray models

3.1 Introduction

From the results of Chapter 2, we know that the 5-equation mixture model is compatible with a model which describes some sub-scale effects, namely bubbles pulsations. The idea of this chapter is to give some clues about a further modeling of microscopic interfacial effects.

Let us recall that the derivation of the equations with micro-inertia effects was performed in Chapter 1 by assuming a pulsation energy, the form of which is inspired from the pulsations of bubbles: the momentum associated with this energy is given by the product of a micro-inertia with the time evolution of the bubbles volumes. Since the pulsation energy is given as a function of mean quantities (like volume fraction), sub-scale quantities standing for the sub-scale structures of the flow do not appear in the final set of equations: the bubbles radii $\mathcal{R}$ and the bubbles density number n are not variables of the two-phase system. In Chapter 2, we have recovered the evolution equations for n and $\mathcal{R}$, through some restrictive assumptions on the sub-scale interface topology:

- bubbles are assumed to be spherical,
- the radius of the bubbles is uniform at a sub-scale level (the repartition in size is **monodisperse**),
- the liquid phase is incompressible in the surroundings of the bubbles.

This leads to a conservation of the bubbles density number (2.8) and a Rayleigh-Plesset-like equation (2.13).

Now, we are interested in more general interfacial structures. In order to use the same procedure as in Chapter 1 and Chapter 2, we first study the geometrical properties of surfaces and their evolution equations. This study is based on the works of Pope (1988); Drew (1990); Morel (2015); Morel (2007). Then, a statistical approach is considered and a Surface Density Function is derived following the lines of Pope (1988). This second stage involves the choice of representative variables of the surface evolution. In a certain extent, this SDF can be related to a Number Density Function (NDF) for the description of a disperse phase (Williams (1958)). Thus, a first link is made with the kinetic methods for polydisperse sprays of droplets and the first fractional moments of the NDF (see Introduction, (10) and Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016); Essadki, de Chaisemartin, Laurent, and Massot (2016)). This study provides interesting information about quantities which could be used in the description of a general two-phase flow, and more specifically in the description of flows of polydisperse and deformed bubbles. This aspect is discussed at the end of this chapter.

Remark 9. *This chapter is based on a current work (Essadki, Drui, de Chaisemartin, Larat, Ménard, and Massot (2017)) in collaboration with IFPEn and CORIA. It is principally based on existing works (Pope (1988); Drew (1990); Morel (2015); Morel (2007)). Results will be presented in Essadki, Drui,*

de Chaisemartin, Larat, Ménard, and Massot (2017) in the context of sprays of droplets and the perspectives of this study will provide concrete research axes for the study of bubbly flows.

3.2 Geometrical description and evolution laws of surfaces

First, we are interested in the description and the evolution of surfaces. We recall here the different ways a surface can be defined. Then, we look for intrinsic properties of the surface, meaning properties which are independent of the choice of the surface definition. These properties evolve according to laws that have been derived in Pope (1988) and in Drew (1990). The section ends with the choice of the quantities that may be used to describe a surface, and will be used as statistical variables.

3.2.1 Geometrical properties of a surface

Let us consider a 3D spatial domain, associated with a fixed referential $(O, \mathbf{x}_1, \mathbf{x}_2, \mathbf{x}_3)$, and a 2D surface $\mathcal{S}(t)$ in this domain. Two definitions for $\mathcal{S}(t)$ are commonly considered:

- an **explicit** definition, based on a couple of parameters $(u, v) \in \mathcal{U}$, $\mathcal{U} \subset \mathbb{R}^2$, where:

$$\begin{array}{rccc} \mathbf{X} & : & \mathcal{U} \times \mathbb{R} & \rightarrow & \mathcal{S}(t) \\ & & (u, v, t) & \mapsto & \mathbf{X}(u, v, t) \end{array}$$

  and

$$\mathcal{S}(t) = \left\{ \mathbf{X}(u, v, t) \in \mathbb{R}^3 \, , \, (u, v) \in \mathcal{U} \right\}, \tag{3.1}$$

- an **implicit** definition, based on an isoline $\phi_0 = 0$ of a function (such as color, level-set functions) $\phi \in \mathrm{C}^2\left(\mathbb{R}^3 \times \mathbb{R}; \mathbb{R}\right)$. Then, the surface is defined by:

$$\mathcal{S}(t) = \{ \mathbf{x} \in \mathbb{R}^3 \, , \, \phi(\mathbf{x}, t) = 0 \}. \tag{3.2}$$

As we will see in the following, the explicit definition has advantages in the definition of the geometrical properties of $\mathcal{S}(t)$, but it assumes that the surface is regular enough (C^2) and cannot handle a set of separated surfaces, which the implicit definition can. Both definitions are arbitrary: the first one depends on the choice of the parametrization (u, v), while the second one depends on the choice of the function ϕ. But, they can both be used to define intrinsic properties of the surface.

Normal to the surface

Let $\mathbf{X}^o(t)$ be a point on the surface at time t such that $\mathbf{X}(u^o, v^o, t) = \mathbf{X}^o(t)$, and $\mathbf{N}^o(t)$ be the unit normal to the surface at $\mathbf{X}^o(t)$. By definition, $\mathbf{N}^o(t)$ is orthogonal to the tangent plane to the surface at $\mathbf{X}^o(t)$. Let us assume that it is oriented in a given direction. Let us consider its expressions according to both surface definitions:

- using the explicit definition of the interface (3.1), $\mathbf{N}^o$ is defined by:

$$\mathbf{N}^o(t) = \left. \frac{\frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v}}{\left| \frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v} \right|} \right|_{(u,v,t)=(u^o,v^o,t)}$$

  where $\times$ denotes the cross product, and $|\cdot|$ the L^2-norm in $\mathbb{R}^3$,

- with the implicit definition of the surface (3.2), we can note that, since ϕ defines a constant-property interface, the derivatives of ϕ along the tangential directions are zero. Thus, $\boldsymbol{\nabla}_{\mathbf{x}} \phi$ is oriented along the direction normal to the tangent plane, and reads (see also Aris (1962); Morel (2007)):

$$\mathbf{N}^o(t) = \left. \frac{\boldsymbol{\nabla}_{\mathbf{x}} \phi}{|\boldsymbol{\nabla}_{\mathbf{x}} \phi|} \right|_{\mathbf{x}=\mathbf{X}^o(t)}.$$

Remark 10. *One can show the equality of these two definitions, by noting that for any* $(u,v) \in \mathcal{U}$, $\phi(\mathbf{X}(u,v,t),t) = 0$. *Then, since* ϕ *is constant (zero) on the surface, one has:*

$$\partial_u \phi = 0 = \frac{\partial \mathbf{X}}{\partial u}^T \boldsymbol{\nabla}_{\mathbf{x}} \phi,$$

$$\partial_v \phi = 0 = \frac{\partial \mathbf{X}}{\partial v}^T \boldsymbol{\nabla}_{\mathbf{x}} \phi.$$

$\boldsymbol{\nabla}_{\mathbf{x}}\phi$ *is orthogonal to* $\partial_u \mathbf{X}$ *and* $\partial_v \mathbf{X}$, *it is then collinear to* $\partial_u \mathbf{X} \times \partial_v \mathbf{X}$. *The norm of the normal vector being unity, it remains to study the orientations of* $\boldsymbol{\nabla}_{\mathbf{x}}\phi$ *and* $\partial_u \mathbf{X}$ *and* $\partial_v \mathbf{X}$. *For the implicit definition of the surface* (3.2), *the orientation depends on the values of function* ϕ *on both sides of the interface. On the other hand, for the explicit definition* (3.1), *the orientation depends on the parametrization* (u,v). *These two choices are arbitrary, and in the following, we can assume that they are compatible, when used to describe a unique piece of surface.*

Surface area density

Let us consider a piece of surface $\mathcal{S}_{\mathcal{V}}(t)$, that is contained in a volume $\mathcal{V}$. With the explicit definition (3.1), the surface area $\mathrm{S}_{\mathcal{V}}(t)$ of $\mathcal{S}_{\mathcal{V}}(t)$ is given by:

$$\mathrm{S}_{\mathcal{V}}(t) = \int_{\mathcal{U}_{\mathcal{V}}} \left| \frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v} \right| \mathrm{d}u\mathrm{d}v,$$

where $\mathcal{U}_{\mathcal{V}}$ is the set of parameters $(u,v) \in \mathcal{U}$ such that $\mathbf{X}(u,v,t) \in \mathcal{V}$. Let us introduce:

$$A(u,v,t) = \left| \frac{\partial \mathbf{X}}{\partial u} \times \frac{\partial \mathbf{X}}{\partial v} \right|.$$

Note that $A(u,v,t)$ is related to the **area density** (or **surface-to-volume ratio**) of the surface $\mathcal{S}(t)$, but depends on the parametrization. Since $A(u,v,t)$ is a regular function over the surface $\mathcal{S}_{\mathcal{V}}(t)$, it naturally defines a distribution on all the test functions of $\mathbb{R}^3$. This way, we introduce a measure Σ of the area density as a distribution function such that:

$$\langle \Sigma, \varphi \rangle = \int_{\mathcal{U}} A(u,v,t) \varphi(\mathbf{X}(u,v,t)) \mathrm{d}u\mathrm{d}v, \tag{3.3}$$

where the notation $\langle \cdot, \cdot \rangle$ is the duality bracket. For the purposes of notations and extensively using the Dirac delta function δ, we write, in the sense of the distributions:

$$\Sigma(t,\mathbf{x}) = \int_{\mathcal{U}} A(u,v,t) \delta(\mathbf{x} - \mathbf{X}(u,v,t)) \mathrm{d}u\mathrm{d}v.$$

Let us underline that the correct mathematical notation is given by (3.3). In particular, we can recover the value of the surface area $\mathrm{S}_{\mathcal{V}}(t)$ by taking $\varphi = \mathbb{1}_{\mathcal{V}}$, the indicator function of $\mathcal{V}$, in (3.3):

$$\mathrm{S}_{\mathcal{V}}(t) = \langle \Sigma, \mathbb{1}_{\mathcal{V}} \rangle .$$

With the second notation, this will write:

$$\mathrm{S}_{\mathcal{V}}(t) = \int_{\mathcal{V}} \Sigma(t,\mathbf{x}) \mathrm{d}\mathbf{x}.$$

With the implicit definition (3.2), the surface area $\mathrm{S}_{\mathcal{V}}(t)$ of the piece of surface $\mathcal{S}_{\mathcal{V}}(t)$ is given by the integral over $\mathcal{S}_{\mathcal{V}}(t)$ of $|\boldsymbol{\nabla}_{\mathbf{x}}\phi|(t,\mathbf{x})$:

$$\mathrm{S}_{\mathcal{V}}(t) = \int_{\mathcal{S}_{\mathcal{V}}(t)} |\boldsymbol{\nabla}_{\mathbf{x}}\phi|(t,\mathbf{x}) \mathrm{d}\mathbf{x},$$

so that, with the same notations as in the previous paragraph, we can write in the sense of the distributions:

$$\Sigma(t,\mathbf{x}) = |\boldsymbol{\nabla}_{\mathbf{x}}\phi| \, \delta(\phi(t,\mathbf{x})). \tag{3.4}$$

Remark 11. *Using the properties of the Dirac delta function, applied on a composed function, we can show that the quantity* $\Sigma(t,\mathbf{x})$ *does not depend on the choice of* ϕ.

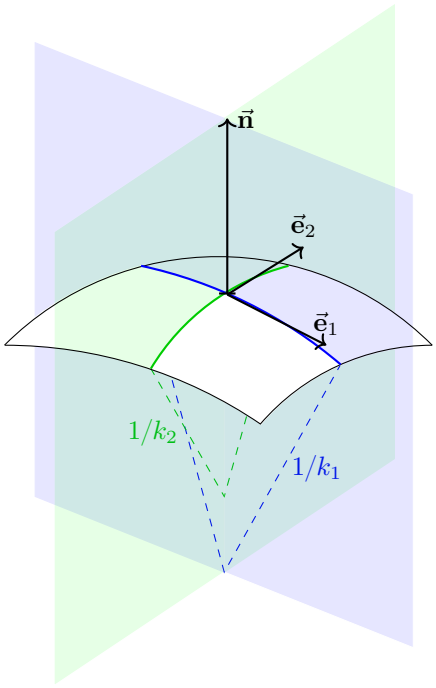


Figure 3.1: *Illustration of the local curvatures at* $\mathbf{X}^o$.

Curvatures

The normal to the interface and the area density may not be sufficient to describe a local piece of surface, as they provide only first-order information: they are related to the first-order derivatives of $\mathbf{X}$ or ϕ. The principal curvatures which are introduced here, are also intrinsic properties of the surface. They provide second-order information.

The definition of these curvatures for the explicit definition of the surface (3.1) follows the lines of Pope (1988). Let us consider $\mathbf{X}^o$, a point of the surface, and a local orthonormal referential associated with $\mathbf{X}^o$, that is $\mathcal{R}^o = (\mathbf{X}^o, \mathbf{e}_1, \mathbf{e}_2, \mathbf{N}^o)$. For any point $\mathbf{X} \in \mathcal{S}$ in the neighborhood of $\mathbf{X}^o$, we know that there exists a height function $h : (y_1, y_2) \mapsto h(y_1, y_2)$ equal to the coordinate of $\mathbf{X}$ along $\mathbf{N}^o$ in the referential $\mathcal{R}^o$, according to the implicit function theorem. Let us also consider y_1, y_2 the coordinates of $\mathbf{X}$ along $\mathbf{e}_1$ and $\mathbf{e}_2$ respectively. Then we have $h(0,0) = 0$ (corresponding to the point $\mathbf{X}^o$), and, if we denote $\frac{\partial h}{\partial y_\alpha}$, $\alpha = 1, 2$ the partial derivative of h along the direction $\mathbf{e}_\alpha$, we have:

$$\left.\frac{\partial h}{\partial y_\alpha}\right|_{\mathbf{X}^o} = 0.$$

Consequently, the first terms of the Taylor series expansion of h at $\mathbf{X}^o$ involve the Hessian matrix $\mathcal{H}_h = \left.\dfrac{\partial^2 h}{\partial y_\alpha \partial y_\beta}\right|_{\mathbf{X}^o}$, $\alpha, \beta \in \{1, 2\}$. If we introduce a 2D vector $\mathbf{y} = (y_1, y_2)$ in the tangent plane, the Taylor expansion reads:

$$h(\mathbf{y}) = \mathbf{y}^T \mathcal{H}_h \mathbf{y} + O(|\mathbf{y}|^3).$$

Being symmetric, $\mathcal{H}_h$ is diagonalizable in an orthogonal basis (Pope (1988)), and its eigenvalues are the **principal curvatures**, k_1, k_2, of the surface at $\mathbf{X}^o$. From a geometric point of view, these curvatures represent the minimum and maximum curvatures at $\mathbf{X}^o$, and if $\mathbf{e}_1^*$ and $\mathbf{e}_2^*$ are the eigenvectors of $\mathcal{H}$, the referential $\mathcal{R}^* = (\mathbf{X}^o, \mathbf{e}_1^*, \mathbf{e}_2^*, \mathbf{N}^o)$ is orthonormal (see an illustration in Figure 3.1). Now, let us introduce the local **mean curvature** H and **Gaussian curvature** G, that are defined by:

$$H = \frac{1}{2}\mathrm{tr}(\mathcal{H}_h) = \frac{1}{2}(k_1 + k_2), \tag{3.5}$$

$$G = \det(\mathcal{H}_h) = k_1 k_2. \tag{3.6}$$

Remark 12. *The application:*

$$\begin{array}{ccc} \mathbb{R}^2 & \to & \mathbb{R}\times\mathbb{R}^+ \\ (k_1,k_2) & \mapsto & (H,G) \qquad s.\ t.\ k_1 \leqslant k_2 \text{ or } k_2 \leqslant k_1, \text{ and } H^2 \geqslant G \end{array}$$

is a bijection.

From the implicit approach (3.2), one can define the Hessian matrix $\mathcal{H}_\phi = \left.\dfrac{\partial^2\phi}{\partial y_\alpha \partial y_\beta}\right|_{\mathbf{X}^o}$, $\alpha,\beta \in \{1,2,3\}$. It can be shown that (Drew and Passman (1999)):

$$\boldsymbol{\nabla}_{\mathbf{x}}\mathbf{N} = -\frac{1}{|\boldsymbol{\nabla}_{\mathbf{x}}\phi|}\left(\mathbf{Id} - \mathbf{N}\mathbf{N}^T\right)\mathcal{H}_\phi.$$

In the referential $\mathcal{R}^*$ defined previously, one has:

$$\boldsymbol{\nabla}_{\mathbf{x}}\mathbf{N} = \begin{pmatrix} k_1 & 0 & l_1 \\ 0 & k_2 & l_2 \\ 0 & 0 & 0 \end{pmatrix} \tag{3.7}$$

where l_1 and l_2 are two real parameters. From this approach, one can also define the mean and Gaussian curvatures using k_1, k_2 and their definitions (3.5) and (3.6).

The normal $\mathbf{N}^o$, the area density measure Σ and the curvatures H and G, provide us all first and second order information about the form of the surface in the vicinity of $\mathbf{X}^o$. Moreover, all these properties are intrinsic properties of the surface. Let us now study how these parameters evolve through time.

3.2.2 Dynamics of a local piece of interface

First let us define a velocity field $\mathbf{v}_I(t,\mathbf{x})$ and the associated Lagrangian derivative of a quantity $a(t,\mathbf{x})$:

$$\dot{a}(t,\mathbf{x}) = \partial_t a(t,\mathbf{x}) + \mathbf{v}_I^T\boldsymbol{\nabla}_{\mathbf{x}}a(t,\mathbf{x}). \tag{3.8}$$

Then, following the works of Pope (1988), Drew (1990); Drew and Passman (1999) and Morel (2007); Morel (2015), we present the evolution laws for the local variables H, G and Σ.

Interfacial velocity

Let us consider $\mathbf{X}(u,v,t)$ a point on the surface $\mathcal{S}(t)$, that stays on the surface through time. The velocity $\mathbf{v}_{\mathcal{U}}(u,v,t)$ of $\mathbf{X}$ along its trajectory for a constant parametrization $\mathcal{U}$ can be defined by:

$$\mathbf{v}_{\mathcal{U}}(u,v,t) = \frac{\partial\mathbf{X}(u,v,t)}{\partial t}.$$

The velocity $\mathbf{v}_{\mathcal{U}}(u,v,t)$ clearly depends on the parametrization (u,v). Let us show however that the application $\mathbf{X}\in\mathcal{S}\mapsto v_I(\mathbf{X},t)$, s. t. $v_I(\mathbf{X}(u,v),t) = \mathbf{v}_{\mathcal{U}}(u,v,t)^T\mathbf{N}(\mathbf{X}(u,v),t)$ does not depend on the choice of the parametrization $\mathcal{U}$. This means that the normal component of the velocity $\mathbf{v}_{\mathcal{U}}$ is uniquely defined.

Let us consider the function $\varphi : (u,v,t)\mapsto\phi(\mathbf{X}(u,v,t),t)$ (ϕ is the same as in (3.2)). By definition, we have $\varphi(u,v,t) = 0$. Then:

$$\frac{\partial\varphi(u,v,t)}{\partial t} = 0.$$

We have $\forall\mathbf{X}\in\mathcal{S}(t)$:

$$\begin{aligned} 0 &= \frac{\partial\phi(\mathbf{X},t)}{\partial t} + \left(\frac{\partial\mathbf{X}(u,v,t)}{\partial t}\right)^T\boldsymbol{\nabla}_{\mathbf{X}}\phi(\mathbf{X},t), \\ &= \frac{\partial\phi(\mathbf{X},t)}{\partial t} + \mathbf{v}_{\mathcal{U}}(u,v,t)^T\boldsymbol{\nabla}_{\mathbf{X}}\phi(\mathbf{X},t), \\ &= \frac{\partial\phi(\mathbf{X},t)}{\partial t} + v_I(\mathbf{X},t)\,|\boldsymbol{\nabla}_{\mathbf{X}}\phi(\mathbf{X},t)| \end{aligned}$$

by recalling that $\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{X},t) = |\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{X},t)|\,\mathbf{N}(\mathbf{X},t)$. Then, $\forall \mathbf{X} \in \mathcal{S}(t)$, $v_I(\mathbf{X},t)$ is given by:

$$v_I(\mathbf{X},t) = -\frac{1}{|\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{X},t)|}\frac{\partial\phi(\mathbf{X},t)}{\partial t}.$$

$v_I(\mathbf{X},t)$ depends on the evolution of function ϕ on the surface, independently of the parametrization $\mathcal{U}$. Moreover, one could show that $v_I(\mathbf{X},t)$ does not depend on the choice of ϕ. Finally, the normal component of the interfacial velocity $(\mathbf{X},t) \mapsto v_I(\mathbf{X},t)$ is uniquely defined.

What about the tangential components? By noticing that for any velocity $\mathbf{v}$, one has:

$$\mathbf{v}^T\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{X},t) = (\mathbf{v}^T\mathbf{N}(\mathbf{X},t)\,|\boldsymbol{\nabla}_{\mathbf{X}}\phi(\mathbf{X},t)|),$$

one can see that the tangential components of the velocity will not influence the evolution of ϕ. Consequently, without loss of generality, we can define the interfacial velocity $\mathbf{V}_I$ for any $\mathbf{X} \in \mathcal{S}$ as:

$$\mathbf{V}_I(\mathbf{X},t) = v_I(\mathbf{X},t)\mathbf{N}(\mathbf{X},t). \tag{3.9}$$

Finally, we can also define a velocity field $\mathbf{v}_I(\mathbf{x},t)$ for any $\mathbf{x} \in \mathbb{R}^3$, such that, when $|\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{X},t)| \neq 0$:

$$\mathbf{v}_I(\mathbf{x},t) = -\frac{\partial\phi(\mathbf{x},t)}{\partial t}\frac{\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{x},t)}{|\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{x},t)|^2}.$$

The function ϕ is then evolved following the following equation:

$$\dot{\phi}(\mathbf{x},t) = \partial_t\phi(\mathbf{x},t) + \mathbf{v}_I(\mathbf{x},t)^T\boldsymbol{\nabla}_{\mathbf{x}}\phi(\mathbf{x},t) = 0. \tag{3.10}$$

Evolution equations for the local variables

We study here the equations that give $\dot{H}$, $\dot{G}$ and $\dot{\Sigma}$ (we recall that it must be treated as the derivative of a distribution). We refer to the works of Drew (1990) for details on the derivation of the first two following equations and to Pope (1988) for the derivation of the equation on Σ. Let us precise here that the definition of Σ as a distribution function was not taken into account in the formalism of Pope (1988). The compatibility of the equations presented below with the definition of Σ is still to be verified. However, let us present the results that are found in these papers, which state that $\forall \mathbf{X} \in \mathcal{S}(t)$:

$$\dot{H}(t,\mathbf{X}) = -\frac{1}{2}\left(\frac{\partial^2 v_I}{\partial {y_1^*}^2} + \frac{\partial^2 v_I}{\partial {y_2^*}^2}\right) - (2H^2 - G)v_I \tag{3.11}$$

$$\dot{G}(t,\mathbf{X}) = -H\left(\frac{\partial^2 v_I}{\partial {y_1^*}^2} + \frac{\partial^2 v_I}{\partial {y_2^*}^2}\right) - 2HGv_I + \sqrt{H^2 - G}\left(\frac{\partial^2 v_I}{\partial {y_1^*}^2} - \frac{\partial^2 v_I}{\partial {y_2^*}^2}\right) \tag{3.12}$$

$$\dot{\Sigma}(t,\mathbf{X}) = 2H\Sigma v_I - \Sigma \mathrm{div}_{\mathbf{x}}\mathbf{v}_I \tag{3.13}$$

One can note that the first terms appearing on the right-hand sides of equations (3.11) and (3.12) are a surface Laplace operator. They do not depend on the choice of the local referential, although they are written here in the referential associated with the principal curvatures. However, the last term appearing in equation (3.12) is given in the local principal directions, which may cause closure issues for the model.

Remark 13. *With $\mathbf{v}_I = v_I\mathbf{N}$, we can note that:*

$$\dot{\mathbf{N}} = \partial_t\mathbf{N} + \mathbf{v}_I^T\boldsymbol{\nabla}_{\mathbf{x}}\mathbf{N} = \partial_t\mathbf{N},$$

and according to Drew and Passman (1999), the equation for $\mathbf{N}$ is then given by:

$$\partial_t\mathbf{N} = -\boldsymbol{\nabla}_{\mathbf{x}}v_I.$$

The system made of equations (3.11), (3.12) and (3.13) is not entirely closed. Indeed, one must also define an equation for $\mathbf{v}_I$ and deal with the different derivatives along the principal directions that are not a Laplacian operator. To deal with these derivatives, some modeling is necessary to replace these terms. While for the interfacial velocity, we could envision to study the balance of forces that are exerted on the surface, like pressure effects, surface tension effects or drag forces, and thus derive a local momentum balance law.

3.2.3 Conclusion on the local description of the interface

To sum up this first section dedicated to the geometrical description of a surface and its evolution, let us recall its main points:

- a surface can be defined in two ways: explicitly (3.1) or implicitly (3.2),
- from both definitions, one can define intrinsic quantities (independent of the surface definition) related to the first and second order surface deformations. These quantities are the normal to the surface, its area density and its principal curvatures, or equivalently its mean and Gaussian curvatures,
- one can define an interfacial velocity field, such that the velocity vector is normal to the surface,
- in the works of Drew (1990) and Pope (1988), one can find evolution equations for the geometric quantities, according to the interfacial velocity field,
- these equations are not closed, and their compatibility with the definition of the interfacial density distribution has not been verified yet.

Let us outline that this description of a surface and the evolution equations of its geometrical properties are considered at a scale where the surface is assumed to be exactly located.

3.3 Surface density function and probabilistic approach

3.3.1 Ensemble averaged variables

The description of the local geometrical parameters presented in the previous section has little interest when one tries to solve the motion of a well-localized interface. In this case, solving equation (3.10) with an appropriate equation on $\mathbf{v}_I$ is sufficient. On the contrary, in situations where strong fluctuations of the flow make the exact localization and tracking of the interface complex, these local geometrical parameters can be used in a statistical approach.

Let us first mention the usual approach that consists in averaging equations (3.11), (3.12) and (3.13) to get the equations for the mean curvatures $\overline{H}$, $\overline{G}$ and mean surface $\overline{\Sigma}$ defined by:

$$\begin{aligned}\overline{H}(t,\mathbf{x}) &= \langle H(t,\mathbf{x})\rangle_{\mathcal{S}}\,,\\ \overline{G}(t,\mathbf{x}) &= \langle G(t,\mathbf{x})\rangle_{\mathcal{S}}\,,\\ \overline{\Sigma}(t,\mathbf{x}) &= \langle \Sigma(t,\mathbf{x})\rangle\,,\end{aligned}$$

where $\langle\cdot\rangle$ now denotes the ensemble average operator and $\langle\cdot\rangle_{\mathcal{S}}$ is the average conditioned upon the presence of the interface:

$$\langle H(t,\mathbf{x})\rangle_{\mathcal{S}} = \frac{\langle H(t,\mathbf{x})\Sigma(t,\mathbf{x})\rangle}{\overline{\Sigma}(t,\mathbf{x})}.$$

$\overline{\Sigma}$ is also called the *expected surface-to-volume ratio*. It is no longer a distribution, since it has been regularized by the averaging procedure.

Using the topological equation of a two-phase medium:

$$\partial_t \boldsymbol{\chi}_k + \mathbf{v}_I^T \boldsymbol{\nabla}\boldsymbol{\chi}_k = 0, \quad k = 1,2,$$

where $\boldsymbol{\chi}_k$ is the characteristic function of phase k, one can also get an equation on the *void fraction* α, which is defined by:

$$\alpha = \langle \boldsymbol{\chi}_1 \rangle\,.$$

Equations for these quantities can be found in Drew (1990); Drew and Passman (1999). These equations may be closed for certain flow configurations, such as bubbly flows.

3.3.2 A probabilistic description of surfaces

Now, let us consider a description of the interface that permits to treat a larger number of statistical quantities. This is the probabilistic approach of Pope (1988): we account for the probability that the surface is in a given *state* at given point $(t, \mathbf{x})$. In the following, this state will be defined by the local values of the curvatures H and G, and by the local value of the interfacial velocity $\mathbf{v}_I$. The other parameters mentioned previously will be given by:

- the normal $\mathbf{N}$ is obtained from the interfacial velocity $\mathbf{v}_I$, since $\mathbf{v}_I = v_I \mathbf{N}$,
- the surface area density will be used as the probability measure through $\overline{\Sigma}$.

Then, at any $(t, \mathbf{x})$, the expected surface-to-volume ratio of the interface, such that its mean and Gaussian curvatures are $H(t, \mathbf{x})$ and $G(t, \mathbf{x})$ respectively, and such that the interfacial velocity is $\mathbf{v}_I(t, \mathbf{x})$ is given by the Surface Density Function (SDF) F_s defined by:

$$F_s(t, \mathbf{x}, \hat{H}, \hat{G}, \hat{\mathbf{v}_I}) = \left\langle \Sigma\, \delta(\hat{H} - H(t, \mathbf{x}))\, \delta(\hat{G} - G(t, \mathbf{x}))\, \delta(\hat{\mathbf{v}_I} - \mathbf{v}_I(t, \mathbf{x})) \right\rangle . \tag{3.14}$$

It is usual to note $\hat{\boldsymbol{\xi}} = (\hat{H}, \hat{G}, \hat{\mathbf{v}_I})$ the phase space, and write F_s simply as:

$$F_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}) = \left\langle \Sigma\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, \mathbf{x})) \right\rangle .$$

The measure $F_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}) d\mathbf{x} d\hat{\boldsymbol{\xi}}$ provides the probable surface area which is contained in the spatial volume $d\mathbf{x}$ around $\mathbf{x}$ and in the phase space volume $d\boldsymbol{\xi}$ around $\boldsymbol{\xi}$.

Now, the mean surface and mean curvatures are given by the first-order moments of F_s. Indeed, one has:

$$\begin{aligned}
\int_{\hat{\boldsymbol{\xi}}} F_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}) d\hat{\boldsymbol{\xi}} &= \int_{\hat{\boldsymbol{\xi}}} < \Sigma\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, \mathbf{x})) > d\hat{\boldsymbol{\xi}} \\
&=< \int_{\hat{\boldsymbol{\xi}}} \Sigma\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, \mathbf{x})) d\hat{\boldsymbol{\xi}} > \\
&=< \Sigma > \\
&= \overline{\Sigma}(t, \mathbf{x})
\end{aligned}$$

and, for instance for the mean *mean* curvature:

$$\begin{aligned}
\int_{\hat{\boldsymbol{\xi}}} \hat{H} F_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}) d\hat{\boldsymbol{\xi}} &= \int_{\hat{\boldsymbol{\xi}}} \hat{H} < \Sigma\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, \mathbf{x})) > d\hat{\boldsymbol{\xi}} \\
&=< \int_{\hat{\boldsymbol{\xi}}} \hat{H} \Sigma\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t, \mathbf{x})) d\hat{\boldsymbol{\xi}} > \\
&=< H(t, \mathbf{x})\, \Sigma > \\
&= \overline{\Sigma}(t, \mathbf{x})\, \overline{H}(t, \mathbf{x})
\end{aligned}$$

This provides a link with the ensemble averaged quantities presented at the beginning of this section. Now, one must also have equations for these quantities: these equations will be derived from the evolution equation for the SDF F_s, called the **generalized population balance equation** (GPBE).

3.3.3 Generalized population balance equation

Let us derive here the evolution equation for F_s. Let us consider the Lagrangian motion of F_s:

$$\dot{F}_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}) = \partial_t F_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}) + \hat{\mathbf{v}_I}^T \boldsymbol{\nabla}_{\mathbf{x}} F_s(t, \mathbf{x}, \hat{\boldsymbol{\xi}}),$$

and let us define the conditional averages $< \cdot >_c$ for any quantity $\boldsymbol{\chi}$ by:

$$\left\langle \boldsymbol{\chi}(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\rangle_c = \frac{\left\langle \boldsymbol{\chi}(t,\mathbf{x},\hat{\boldsymbol{\xi}})\, \Sigma\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) \right\rangle}{F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})}.$$

Now, we have from the definition of F_s (3.14):

$$\begin{aligned}
\partial_t F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) &=< \{\partial_t \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) + \Sigma \left\{ \partial_t \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) \right\} > \\
&=< \{\partial_t \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > - < \Sigma \partial_t \boldsymbol{\xi}(t,\mathbf{x}) \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&=< \{\partial_t \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > - \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} < \Sigma \partial_t \boldsymbol{\xi}(t,\mathbf{x}) \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&=< \{\partial_t \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > - \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} \left\{ \langle \partial_t \boldsymbol{\xi}(t,\mathbf{x}) \rangle_c\, F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\}
\end{aligned}$$

and

$$\begin{aligned}
\hat{\mathbf{v}_I}^T \boldsymbol{\nabla}_{\mathbf{x}} F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) &=< \hat{\mathbf{v}_I}^T \{\boldsymbol{\nabla}_{\mathbf{x}} \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) + \Sigma \left\{ \hat{\mathbf{v}_I}^T \boldsymbol{\nabla}_{\mathbf{x}} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) \right\} > \\
&=< \hat{\mathbf{v}_I}^T \{\boldsymbol{\nabla}_{\mathbf{x}} \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) + \Sigma \left\{ \mathbf{v}_I(t,\mathbf{x})^T \boldsymbol{\nabla}_{\mathbf{x}} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) \right\} \\
&\quad + \Sigma \{\mathrm{div}_{\mathbf{x}} \mathbf{v}_I(t,\mathbf{x})\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&=< \mathbf{v}_I(t,\mathbf{x})^T \{\boldsymbol{\nabla}_{\mathbf{x}} \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > - < \Sigma \mathbf{v}_I(t,\mathbf{x})^T \boldsymbol{\nabla}_{\mathbf{x}} \boldsymbol{\xi}(t,\mathbf{x}) \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&\quad + < \Sigma \{\mathrm{div}_{\mathbf{x}} \mathbf{v}_I(t,\mathbf{x})\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&=< \mathbf{v}_I(t,\mathbf{x})^T \{\boldsymbol{\nabla}_{\mathbf{x}} \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > - \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} < \Sigma \mathbf{v}_I(t,\mathbf{x})^T \boldsymbol{\nabla}_{\mathbf{x}} \boldsymbol{\xi}(t,\mathbf{x}) \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&\quad + < \Sigma \{\mathrm{div}_{\mathbf{x}} \mathbf{v}_I(t,\mathbf{x})\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > \\
&=< \mathbf{v}_I(t,\mathbf{x})^T \{\boldsymbol{\nabla}_{\mathbf{x}} \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > - \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} \left\{ \left\langle \mathbf{v}_I(t,\mathbf{x})^T \boldsymbol{\nabla}_{\mathbf{x}} \boldsymbol{\xi}(t,\mathbf{x}) \right\rangle_c F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\} \\
&\quad + \langle \mathrm{div}_{\mathbf{x}} \mathbf{v}_I(t,\mathbf{x}) \rangle_c\, F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})
\end{aligned}$$

In the transition from the first to second line, we have used the following development:

$$\begin{aligned}
\hat{\mathbf{v}_I}^T \boldsymbol{\nabla}_{\mathbf{x}} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) &= \mathrm{div}_{\mathbf{x}} \left\{ \hat{\mathbf{v}_I} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) \right\} \\
&= \mathrm{div}_{\mathbf{x}} \left\{ \mathbf{v}_I(t,\mathbf{x}) \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) \right\} \\
&= \mathbf{v}_I(t,\mathbf{x})^T \boldsymbol{\nabla}_{\mathbf{x}} \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) + \{\mathrm{div}_{\mathbf{x}} \mathbf{v}_I(t,\mathbf{x})\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x}))
\end{aligned}$$

By combining all these terms, we find the following balance equation:

$$\partial_t F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) + \mathrm{div}_{\mathbf{x}} \left\{ \hat{\mathbf{v}_I} F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\} + \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} \left\{ \left\langle \dot{\boldsymbol{\xi}}(t,\mathbf{x}) \right\rangle_c F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\} =< \{\partial_t \Sigma + \mathbf{v}_I(t,\mathbf{x})^T \boldsymbol{\nabla}_{\mathbf{x}} \Sigma\}\, \delta(\hat{\boldsymbol{\xi}} - \boldsymbol{\xi}(t,\mathbf{x})) > .$$

By using (3.13), one can finally write the balance equation as:

$$\partial_t F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) + \mathrm{div}_{\mathbf{x}} \left\{ \hat{\mathbf{v}_I} F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\} + \boldsymbol{\nabla}_{\hat{\boldsymbol{\xi}}} \left\{ \left\langle \dot{\boldsymbol{\xi}}(t,\mathbf{x}) \right\rangle_c F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}) \right\} = \langle 2Hv_I - \mathrm{div}_{\mathbf{x}} \mathbf{v}_I \rangle_c\, F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}}). \quad (3.15)$$

We can note that the right-hand side of the balance equation (3.15) is a stretching source term.

Much work has still to be done to close the system of equations, like finding closure laws for $\dot{\mathbf{v}_I}$ or modeling the conditional averaged terms in $< \cdot >_c$, and maybe considering appropriate assumptions to reduce the number of variables in the phase space $\boldsymbol{\xi}$. Considering the different works of Drew and Passman (1999), this can be done by the equations on the first-order moments of F_s: these equations may be related with the equations for the mean geometrical variables $\overline{H}$, $\overline{G}$ and Σ.

In particular, one of the situation where simplifications may occur is when F_s is used to describe a disperse-phase flow made of spherical elements (droplets for instance). We present in the next section how the surface density function F_s can degenerate to an Number Density Function in that precise case.

3.4 Surface density function in the disperse phase flow limit

3.4.1 Definition of the number density function

Presented in the Introduction of this thesis, a disperse-phase flow can be described by using a kinetic approach, based on a number density function (NDF) F_N and Williams-Boltzman equation. When thermal effects are neglected, the phase space $\hat{\boldsymbol{\xi}}_N$ for the NDF is made of shape parameters of the particles (it is often its surface S when assuming spherical particles), and a parameter on its mass center velocity $\mathbf{u}$.

The measure $F_N(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N)\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}}_N$ gives the probable number of particles contained in the spatial volume $\mathrm{d}\mathbf{x}$ around $\mathbf{x}$ and in the phase space volume $\mathrm{d}\boldsymbol{\xi}_N$ around $\boldsymbol{\xi}_N$. Considering that the particles are punctual objects (they are located at their mass center), and considering a flow made of N particles, the Number Density Function is defined as:

$$F_N(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N) = \left\langle \sum_{q=1}^{N} \delta(\mathbf{x}-\mathbf{x}_q)\delta(\hat{\boldsymbol{\xi}}_N - \boldsymbol{\xi}_{N,q}) \right\rangle,$$

Let us assume that the internal variables of the particles are: $\boldsymbol{\xi}_{N,q} = \left(\overline{H}^p_q, \overline{G}^p_q, V_q, \mathbf{u}_q\right)$. Here $\overline{H}^p_q$ is the average of the mean curvature over the particle real surface (weighted by the surface area). For a particle q, located at $\mathbf{x}_q$ and whose surface is $\mathcal{S}_q(t,\mathbf{x}_q)$ of area $\Sigma_q(t,\mathbf{x}_q)$, it is defined by:

$$\overline{H}^p_q(t,\mathbf{x}_q) = \left\langle \Sigma(t,\mathbf{x}), H(t,\mathbf{x})\mathbb{1}_{\mathcal{V}_q} \right\rangle$$

where $\mathcal{V}_q$ is a spatial volume containing the entire particle q and no other particle. In the same way, $\overline{G}^p_q$ is the average of the Gaussian curvature over the particle surface. V_q is the particle volume. About the velocity of the particle, one can note that the surface averaging provides a mean velocity $\overline{\mathbf{v}_I}^p_q$ that could be different from the velocity of the mass center $\mathbf{u}_q$ due to surface deformations. This question must be studied in future works, and for now on, we make the assumption that $\overline{\mathbf{v}_I}^p_q \approx \mathbf{u}_q$. These internal properties of the particles can be used to describe a flow made of deformed particles.

With the assumption that all particles are spherical, one can easily define a simplified version of the NDF. Indeed, one has for a particle q:

$$\begin{aligned}
\overline{H}^p_q &= H(\overline{G}^p_q) = (\overline{G}^p_q)^{-1/2}, \\
V_q &= V(\overline{G}^p_q) = \frac{4}{3}\pi(\overline{G}^p_q)^{3/2}.
\end{aligned}$$

Finally, let us introduce the NDF $n(t,\mathbf{x},\overline{G}^p_q,\mathbf{u})$ in the case of spherical bubbles defined by:

$$F_N(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N) = n(t,\mathbf{x},\widehat{\overline{G}^p},\hat{\mathbf{u}})\delta(\widehat{\overline{H}^p} - H(\widehat{\overline{G}^p}))\delta(\hat{V} - V(\widehat{\overline{G}^p}))$$

3.4.2 Discrete Surface Density Function

Without making the strong assumption of spherical particles, let us just assume that the particles are homotopic to spheres. We can then use the Gauss-Bonnet theorem (Weisstein (2017)), to show that for any such particle q, its surface area Σ_q writes:

$$\Sigma_q = \frac{4\pi}{\overline{G}^p_q}.$$

From the definition of F_N, we can now write:

$$\begin{aligned} F_N(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N) &= \left\langle \sum_{q=1}^{N} \delta(\mathbf{x}-\mathbf{x}_q)\delta(\hat{\boldsymbol{\xi}}_N - \boldsymbol{\xi}_{N,q}) \right\rangle \\ &= \frac{1}{4\pi}\left\langle \sum_{i}^{N} \Sigma_{q=1}\overline{G}^p_q \delta(\mathbf{x}-\mathbf{x}_q)\delta(\hat{\boldsymbol{\xi}}_N - \boldsymbol{\xi}_{N,q}) \right\rangle \\ &= \frac{1}{4\pi}\left\langle \sum_{i}^{N} \Sigma_{q=1}\widehat{\overline{G}^p} \delta(\mathbf{x}-\mathbf{x}_q)\delta(\hat{\boldsymbol{\xi}}_N - \boldsymbol{\xi}_{N,q}) \right\rangle \\ &= \frac{\widehat{\overline{G}^p}}{4\pi} F_p(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N). \end{aligned} \tag{3.16}$$

F_p is a new Surface Density Function, since it has the dimension of a surface density (the invert of $\overline{G}^p$). It is called the Discrete Surface Density Function.

The final results relates the first-order moments of F_p to the first-order moments of F_s. Let us consider a volume $\mathcal{V}$, containing N particles, such that the border of $\mathcal{V}$ does not cross any particle. We have then the following equalities:

$$\int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}\in\Omega_{\boldsymbol{\xi}}} F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}} = \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}_N\in\Omega_{\overline{\boldsymbol{\xi}}^p}} F_p(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N)\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}}_N, \tag{3.17}$$

$$\int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}\in\Omega_{\boldsymbol{\xi}}} \hat{H}F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}} = \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}_N\in\Omega_{\overline{\boldsymbol{\xi}}^p}} \widehat{\overline{H}^p}F_p(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N)\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}}_N, \tag{3.18}$$

$$\int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}\in\Omega_{\boldsymbol{\xi}}} \hat{G}F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}} = \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}_N\in\Omega_{\overline{\boldsymbol{\xi}}^p}} \widehat{\overline{G}^p}F_p(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N)\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}}_N, \tag{3.19}$$

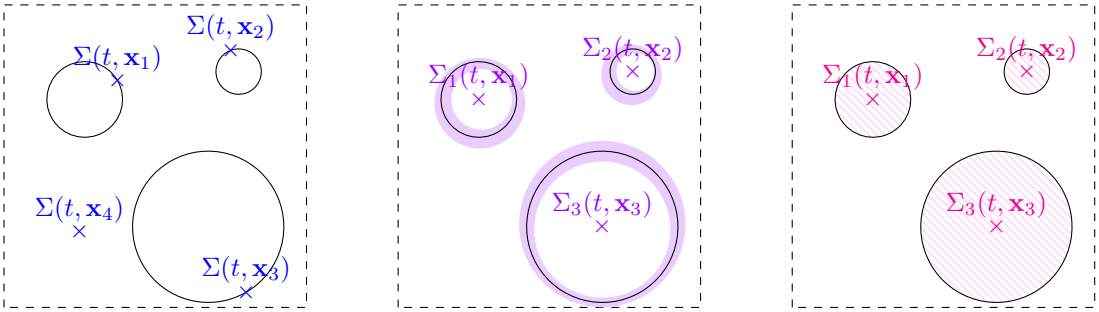
$$\int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}\in\Omega_{\boldsymbol{\xi}}} \hat{\mathbf{v}}_I F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}} = \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}_N\in\Omega_{\overline{\boldsymbol{\xi}}^p}} \widehat{\overline{\mathbf{v}_I}^p}F_p(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N)\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}}_N. \tag{3.20}$$

$$\tag{3.21}$$

Let us show this property for the first-order moment in $\hat{H}$:

$$\begin{aligned} \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}\in\Omega_{\boldsymbol{\xi}}} \hat{H}F_s(t,\mathbf{x},\hat{\boldsymbol{\xi}})\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}} &= \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}\in\Omega_{\boldsymbol{\xi}}} \hat{H}\left\langle \Sigma\,\delta(\hat{\boldsymbol{\xi}}-\boldsymbol{\xi}(t,\mathbf{x}))\right\rangle \mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}} \\ &= \left\langle \int_{\mathbf{x}\in\mathcal{V}} H(t,\mathbf{x})\Sigma\mathrm{d}\mathbf{x}\right\rangle \\ &= \left\langle \sum_{q=1}^{N}\int_{\mathcal{V}_q} H(t,\mathbf{x})\Sigma\mathrm{d}\mathbf{x}\right\rangle \\ &= \left\langle \sum_{q=1}^{N} \Sigma_q \overline{H}^p_q \right\rangle \\ &= \left\langle \sum_{q=1}^{N}\int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}_N\in\Omega_{\boldsymbol{\xi}_n}} \Sigma_q\overline{H}^p_q\delta(\hat{\boldsymbol{\xi}}-\hat{\boldsymbol{\xi}}_{N,q})\delta(\mathbf{x}-\mathbf{x}_q)\mathrm{d}\hat{\boldsymbol{\xi}}\mathrm{d}\mathbf{x}\right\rangle \\ &= \int_{\mathbf{x}\in\mathcal{V}}\int_{\hat{\boldsymbol{\xi}}_N\in\Omega_{\boldsymbol{\xi}_n}} \widehat{\overline{H}^p}F_p(t,\mathbf{x},\hat{\boldsymbol{\xi}}_N)\mathrm{d}\mathbf{x}\mathrm{d}\hat{\boldsymbol{\xi}}_N \end{aligned}$$

The same demonstration applies to the zeroth-order and other first-order moments of F_s and F_p. These zeroth- and first-order moments are called the **topological moments**.



(a) Surface Density Function statistics (Σ distribution function) evaluated locally on the surface.

(b) Mean Surface Density Function statistics (Σ_i particles surface area): averages over volumes containing one entire particle.

(c) Number Density Function statistics (Σ_i internal variable) associated with the mass center of the drops.

Figure 3.2: *Illustration of one realization of a set of three drops and what the different statistics stand for.*

3.4.3 Towards a relation between the Surface Density Function and the Number Density Function

Let us sum up the different distribution functions defined in the previous paragraphs (see also a schematic representation in Figure 3.2):

- the Surface Density Function F_s provides the statistics of local properties of a surface, or a set of surfaces. These local properties are the mean curvature H, the Gaussian curvature G and the interface velocity $\mathbf{v}_I$,
- the discrete Surface Density Function F_p provides the statistics of averaged properties of a set of surfaces associated with a set of particles. The averaging is made on each particle and provides a mean *mean* curvature $\overline{H}^p$, a mean Gaussian curvature $\overline{G}^p$ and the mean interfacial velocity $\overline{\mathbf{v}_I}^p$, which is assumed to be close to the velocity of the mass center of the particle,
- the Number Density Function F_N provides the statistics of a set of particles, for which the surface (related to the mean Gaussian curvature according to the Gauss-Bonnet theorem), the mean *mean* curvature and the velocity of its mass center are internal properties.

In a specific configuration, where we assume a disperse-phase flow and a spatial volume $\mathcal{V}$ such that in all the realizations of the flow, the borders of $\mathcal{V}$ do not cross any particle, we have also related the volume-averaged first-order moments of F_p and F_s (relations (3.17) to (3.20)). The next steps for a full relation between F_s and F_p will necessitate to define a volume-averaging strategy, which preserves the topological moments and which can be applied to any flow configuration (not only for disperse-phase flows). This averaging procedure will further necessitate to establish accurate closure relations.

Finally, we have shown how F_N and F_p are related (equation (3.16)). Using the Gauss-Bonnet theorem and F_p, it seems possible to describe flows of deformed drops. However, these types of flows have not been studied yet from a statistical point of view. Only the spherical case is well known, where the fractional moments of the NDF are used to describe the polydispersion in size of the droplets (Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016)).

3.5 Perspectives for the description of bubbly flows

The work presented in the previous sections is still under development in Essadki, Drui, de Chaisemartin, Larat, Ménard, and Massot (2017), so as to provide answers to the remaining issues on:

- how to close the equations for the dynamics of the geometrical variables: H, G, Σ, $\mathbf{v}_I$ and for the averaged quantities $\overline{H}$, $\overline{G}$, $\overline{\Sigma}$, $\overline{\mathbf{v}_I}$,

- how to properly average the SDF, F_s, to recover the discrete SDF F_p for any flow configuration,
- what is the physical meaning of the first-order moments of F_p in a case of a disperse flow made of deformed droplets.

Up to now, it seems to show that the mean geometrical quantities $\overline{H}$ and $\overline{G}$ play an important role, since they are related to the first-order moments of surface distribution functions and are also mean quantities of the two-phase flow. In the context of bubbly flows, it could be interesting to consider energies related to these quantities for deformed bubbles. Indeed, the bubble deformation may be characterized by the values of $\overline{H}$ and $\overline{G}$. For example, a surface potential energy could be written in function of these quantities. In the same way, the pulsation energy could apply to deformed bubbles and then, the volume variations would be functions of the geometrical parameters. Departing from these energies and using the procedure based on Hamilton's variational principle and the second principle of thermodynamics, as presented in Chapter 1, we may enrich again the two-phase models, and get closer relations between the mixture models and the kinetic-based models.

Part II

Discretization strategy for two-phase mixture models

Introduction

Recall of the systems of equations and their mathematical properties

In the following chapters, we propose a discretization strategy to solve the three two-phase models derived in Chapter 1, section 1.4, numerically. First, let us recall that the systems of equations are:

- the 3-equation homogeneous equilibrium model:

$$\left\{\begin{array}{rcrcl} \partial_t \rho & + & \operatorname{div}(\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho Y) & + & \operatorname{div}(\rho Y \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \operatorname{div}\left(\rho \mathbf{u}\mathbf{u}^T + p\,\mathbf{Id}\right) & = & 0 \end{array}\right.$$
$$p = p_1(\rho, Y) = p_2(\rho, Y)$$

- the 4-equation model with relaxation:

$$\left\{\begin{array}{rcrcl} \partial_t \rho & + & \operatorname{div}(\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho Y) & + & \operatorname{div}(\rho Y \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \operatorname{div}\left(\rho \mathbf{u}\mathbf{u}^T + p\,\mathbf{Id}\right) & = & 0 \\ \partial_t \alpha & + & \mathbf{u}^T \boldsymbol{\nabla} \alpha & = & \dfrac{p_1 - p_2}{\epsilon} \end{array}\right.$$
$$p = \alpha p_1(\rho, Y, \alpha) + (1-\alpha) p_2(\rho, Y, \alpha)$$

- and the 5-equation system with micro-inertia and micro-viscosity:

$$\left\{\begin{array}{rcrcl} \partial_t \rho & + & \operatorname{div}(\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho Y) & + & \operatorname{div}(\rho Y \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \operatorname{div}\left(\rho \mathbf{u}\mathbf{u}^T + \left(p + \frac{1}{2}(\rho Y w)^2\right)\,\mathbf{Id}\right) & = & 0 \\ \partial_t \alpha & + & \mathbf{u}^T \boldsymbol{\nabla} \alpha & = & \dfrac{\rho Y w}{\sqrt{\nu}} \\ \partial_t w & + & \mathbf{u}^T \boldsymbol{\nabla} w & = & \dfrac{p_1 - p_2}{\rho Y \sqrt{\nu}} - \dfrac{\epsilon}{\nu} w \end{array}\right.$$
$$p = \alpha p_1(\rho, Y, \alpha) + (1-\alpha) p_2(\rho, Y, \alpha)$$

These systems are associated with barotropic pressure laws $p_k(\rho_k)$, $k = 1, 2$ with $\rho_1 = (\rho Y)/\alpha$ and $\rho_2 = (\rho(1-Y))/(1-\alpha)$ such that $\partial_{\rho_k} p_k = c_k^2 > 0$ (the notation $\partial_\rho p = \partial p / \partial \rho$ will be used in the following). The convex set $\boldsymbol{\Omega}$ of admissible values for this system is defined as follows:

$$\boldsymbol{\Omega} = \left\{ (\rho, Y, \alpha)^T \in \mathbb{R}^3 \;\middle|\; \begin{array}{c} \rho > 0 \\ Y \in]0,1[\\ \alpha \in]0,1[\end{array} \right\}.$$

Remark 14. *For the 3-equation system, the values $Y = 0, 1$ and $\alpha = 0, 1$ are also handled by the model.*

Generic form of the equations: in the following and according to the **rotational invariance** of all the three systems (all directions are solved in the same way), it is convenient to write the three two-phase systems in 1D under the following form:

$$\left\{\begin{array}{rcrcl} \partial_t (\rho u) & + & \partial_x \left(\rho u^2 + P\right) & = & 0 \\ \partial_t \rho & + & \partial_x (\rho u) & = & 0 \\ \partial_t \mathbf{V} & + & u \partial_x \mathbf{V} & = & \mathbf{S}(\rho, \mathbf{V}) \end{array}\right. \tag{4.22}$$

where $\mathbf{V}$ is a vector of transported variables. $\mathbf{V}$ contains the two other components of the velocity field u_y and u_z, the mass fraction Y and according to the two-phase model, the volume fraction α and/or w. $P(\rho, \mathbf{V})$ is a pressure function, such that

$$\partial_\rho P(\rho, \mathbf{V}) = c^2 > 0, \quad \forall (\rho, \mathbf{V}) \in \mathbf{\Omega}. \tag{4.23}$$

c is the characteristic velocity, which depends on the two-phase system. Let us also assume:

$$\frac{\partial c}{\partial \rho} \geqslant 0. \tag{4.24}$$

Finally $\mathbf{S}(\rho, \mathbf{V})$ stands for the source terms.

Mathematical properties: now, let us give some mathematical properties of system (4.22), that can help design an appropriate numerical scheme. The convective part of system (4.22) is hyperbolic, under the assumptions (4.23) and (4.24). Indeed, we show in Appendix A that the associated matrix obtained in the weak formulation of (4.22), is diagonalizable and possesses three real eigenvalues that are:

$$\lambda_1 = u - c\,, \quad \lambda_2 = u + c\,, \quad \lambda_3 = u\,.$$

We also show in Appendix A that the associated eigenvectors are linearly independent and that:

- the fields associated with λ_1 and λ_2 are genuinely non linear,
- the fields associated with λ_3 are linearly degenerate.

The waves associated with λ_1 and λ_2 are thus shocks or rarefaction waves, while the wave associated with λ_3 is a contact discontinuity. For each of the three two-phase systems, it is possible to compute the exact entropic solution to the Riemann problems (this is done in Appendix A). However, this exact resolution may be complex, especially for the 3-equation system, where the solution is obtained only through the resolution of an implicit relation. Consequently, the numerical methods presented in the following, and in Chapter 4 dedicated to finite volume methods, will rely on approximate solutions of the Riemann problems. Indeed, for all choices we make concerning the discretization strategy, we will look for a balance between the accuracy of the method and its computational cost, while ensuring its robustness.

Review of numerical methods for the two-phase systems

These three systems, or close forms of equations, are encountered in different works on two-phase flow simulations. Let us give a (non-exhaustive) list of these works, and their associated numerical schemes.

3-equation model

The 3-equation system is used Chanteperdrix, Villedieu, and Vila (2002) for simulations of sloshing. The equations are solved with first-order and second-order finite volume schemes, using a relaxation procedure (Jin and Xin (1995)). This procedure consists in solving the equations of the 4-equation system with an exact Godunov scheme (Godunov (2008)) and then projecting the solution on its equilibrium state: the volume fraction variable α is computed so as to satisfy the mechanical equilibrium $p_1 = p_2$, and the pressure is then set as $p = p_1 = p_2$. The second-order scheme is obtained with a MUSCL (van Leer (1979)) reconstruction and a second-order Runge-Kutta time integration. Both methods show good results for classic Riemann problems, and for the sloshing tests with artificially low sound velocities c_k.

In Grenier, Vila, and Villedieu (2013), the 3-equation system is solved with an upwind finite volume scheme and a centered pressure discretization. The scheme is implicit for the mass equations and explicit with a stabilizing term for the momentum equation. These choices are made so as to avoid problems with low-Mach number flows. The tests of sloshing presented in Grenier, Vila, and Villedieu (2013) do not require an artificial lowering of the sound velocities. These tests highlight the issues related to the low-Mach number regime and also show that the choice of the type of mesh (unstructured or cartesian)

has a non-negligible influence: the resolution was better on unstructured grids than on cartesian grids. This is also our experience.

Still with the objective of performing simulations in the low-Mach number regime, in Bernard-Champmartin and De Vuyst (2014), the 3-equation system was discretized on a staggered grid: in the momentum equation, pressure is taken at the centers of the cells, while velocity is given at their edges. The method also uses the low-diffusive approach from Després and Lagoutière (1999); Després and Lagoutière (2001); Kokh and Lagoutière (2010): a downwind discretization of the variable Y is used when possible, *i.e.* when this does not cause negative density values. The simulations of dam break problems and their comparison with experiments show that the method provides physically correct results with a decrease of the diffusion near the two-phase interface.

4-equation model

In Chanteperdrix, Villedieu, and Vila (2002), the 4-equation system is used as an auxiliary system within a relaxation procedure used to solve the 3-equation system, as mentioned previously. Indeed, for this relaxation system, the entropic solution of Riemann problems can be computed exactly with simple explicit relations, that are given in Chanteperdrix, Villedieu, and Vila (2002) (see also in Appendix A). Thus, an exact Godunov method can be used to compute the numerical fluxes associated with the 4-equation system. However, the physics related to this model, where the relaxation source term may play a significant role, is not studied in Chanteperdrix, Villedieu, and Vila (2002), since the source term is not integrated and the values of the variables are forced to their equilibrium state ($p_1 = p_2$).

A system close to the 4-equation is studied and used in Allaire, Clerc, and Kokh (2002); Kokh and Lagoutière (2010): its differences are the presence of an energy equation and no source term in the equation on α (which is moreover considered as a color function). As in Chanteperdrix, Villedieu, and Vila (2002), the physics which is modeled is in fact the physics of the 3-equation system, since mechanical equilibrium is imposed: $p_1 = p_2$. The system is solved numerically using a Lagrange-Remap strategy (Godlewski and Raviart (1996)), where the acoustic and the transport parts of the equations are solved in two different steps. The acoustic part is solved in Lagrangian coordinates, using a Godunov-type solver. The transport part is then solved with the low-diffusive procedure (Després and Lagoutière (1999); Després and Lagoutière (2001); Kokh and Lagoutière (2010)). Numerical tests show an increase of accuracy of the method comparing with a standard upwind discretization of the color function.

Most of the approaches using the 4-equation system, or systems using a pressure relaxation source term (like the Baer-Nunziato system of Baer and Nunziato (1986)), do not integrate the source term. Rather, the method involves a projection of the solution on its mechanical equilibrium state (see the scheme proposed in Saurel and Abgrall (1999), which is widely used for two-phase flow simulations).

5-equation model

The 5-equation homogeneous system with micro-inertia is close to the multiphase system of Gavrilyuk and Saurel (2002): this last system takes two velocities into account, as well as equations on energies. The associated numerical scheme is adapted from Saurel and Abgrall (1999): it consists in a HLL flux for the conservative terms and specific discretization for the non-conservative ones. In particular, an upwind discretization is used for the transport equation of α. In Gavrilyuk and Saurel (2002), it is mentioned that a classic source term integration method is used, but no further details are given.

Summary

Let us sum up the different aspects and issues addressed (or not) in this short review:

1. most of the numerical schemes proposed are based on standard finite volume approaches with Godunov-type schemes: since the models are compressible, shock may occur and must be handled by the numerical method,
2. the equation on the volume fraction α in the 4-equation and 5-equation systems necessitates a particular discretization,

3. the source term of the 4-equation system is rarely integrated as a relaxation source term. In fact, the 4-equation system is often used as an auxiliary system in a relaxation procedure,
4. the compressibility of the models may have side effects in a low-Mach number flow regime: dedicated numerical methods must be used. Moreover, the spatial discretization seems to influence the accuracy of the solution in the low-Mach number regime,
5. the two-phase interface is spread due to numerical diffusion. Improvement of the methods, like low-diffusive schemes, refined meshes or higher-order methods, can be used to reduce this diffusion.

We are addressing these issues within the development of our strategy, presented below.

Operator splitting methods

Let us consider system (4.22) and note that it can be written in the form (it is not unique):

$$\partial_t \mathbf{W} + \mathcal{A}\mathbf{W} - \mathcal{B}\mathbf{W} = 0, \tag{4.25}$$

where $\mathcal{A}$ stands for the convective operators and $\mathcal{B}$ stands for the source terms.

The first aspect of our numerical strategy is an **operator splitting** or **fractional step method** (first uses of the method in Marchuk (1968); Strang (1968); Téman (1969a); Téman (1969b), see also Duarte (2012) and references therein), that enables to solve the parts of the equations associated with the different operators in separate steps. For (4.25), and with an initial condition $\mathbf{W}(t=0,\mathbf{x}) = \mathbf{W}^{(0)}(\mathbf{x})$, this may come down to solve successively the two following problems from time $t = 0$ to time $t = t_f$ (this corresponds to Lie splitting, Trotter (1959)):

$$\begin{cases} \partial_t \mathbf{W}^{(1)}(t,\mathbf{x}) + \mathcal{A}\mathbf{W}^{(1)}(t,\mathbf{x}) = \mathbf{0}, \\ \mathbf{W}^{(1)}(0,\mathbf{x}) = \mathbf{W}^{(0)}(\mathbf{x}), \end{cases} \tag{4.26a}$$

$$\begin{cases} \partial_t \mathbf{W}^{(2)}(t,\mathbf{x}) - \mathcal{B}\mathbf{W}^{(2)}(t,\mathbf{x}) = \mathbf{0}, \\ \mathbf{W}^{(2)}(0,\mathbf{x}) = \mathbf{W}^{(1)}(t_f,\mathbf{x}). \end{cases} \tag{4.26b}$$

Operator splitting methods are widely used, since they facilitate the resolution of complex systems. Indeed, the resolution of each sub-step can be performed with appropriate methods: for example, standard Godunov-type schemes for the convective part, Ordinary Differential Equations (ODE) solvers for the source term, etc. and these methods can be of high-order accuracy. Moreover, when one operator is much stiffer than the others and necessitates a much smaller time step for being accurately integrated, operator splitting allows time-stepping or any other appropriate integration method, since each part of the equation is evolved independently. In Chalons, Girardin, and Kokh (2013); Chalons, Kestener, Kokh, and Stauffert (2016) for example, the acoustic part and the transport part of the equations are split (according to the Lagrange-Projection procedure of Godlewski and Raviart (1996)): the acoustic part is solved implicitly, so as to avoid the CFL constraint on the acoustic waves, while the transport part is solved with an explicit upwind scheme.

However, some care must be taken when using operator splitting methods, and many studies have been conducted on different difficulties raised by these methods.

First, operator splitting may introduce a numerical error, called the *splitting error*. It is given by the difference between the solutions obtained with and without operator splitting: we refer to Leveque (2004); Duarte (2012) where this problem is clearly explained in the linear case and some notions are given for the non-linear case (see also Descombes, Duarte, Dumont, Laurent, Louvet, and Massot (2014)). For the equation (4.25), one can show that, if the operators $\mathcal{A}$ and $\mathcal{B}$ commute ($\mathcal{AB} = \mathcal{BA}$), there will be no splitting error. But they do not commute in the general case and Lie splitting procedure (4.26) may generate a truncation error of the order of the time step of integration Δt. Then, the whole scheme degenerates to a first-order scheme, whatever the order of accuracy of the methods considered for the integration of the different operators. A second order operator splitting is obtained by using Strang splitting (Strang (1968)), where (4.26a) is first integrated over $\Delta t/2$, then (4.26b) is integrated over Δt

and finally (4.26a) is again integrated over $\Delta t/2$. In Descombes, Duarte, Dumont, Laurent, Louvet, and Massot (2014), it is moreover shown that it is preferable to integrate the stiffest operator in the last step. A same order of accuracy may be obtained with the Lie operator splitting, when (4.26a) and (4.26b) are solved alternatively in first or second positions from one time iteration to the other (Leveque (2004)).

Another delicate point concerns boundary conditions, which may not be consistent with the whole coupled problem. This issue will not be discussed in our work, and we refer to Hundsdorfer and Verwer (2003) for examples and discussions.

One may also encounter difficulties with steady-state solutions. These solutions may be hardly solved when using the fractional step method, when integrating independently the convective part and the source terms of the equations. This so-called *well-balanced* issue has been studied and solutions have been proposed in many works (Cargo and le Roux (1994); Greenberg and Roux (1994); Gosse and Roux (1996); Gosse (2000); Chinnayya, Roux, and Seguin (2004); Audusse, Bouchut, Bristeau, Klein, and Perthame (2004)). We propose to adapt one of them for the 3-equation system with a gravity source term and in a context of Adaptive Mesh Refinement (AMR) in Appendix D.

When the source terms are stiff, for example when $\epsilon \to 0$ in the 4-equation system and that the convective time step (computed according to a CFL condition) is much larger than the characteristic time of the relaxation effects, an error due to the splitting strategy may cause an uncontrolled increase of the numerical diffusion. This is especially the case when, in the limit of infinitely fast relaxation towards equilibrium, the asymptotic equilibrium system has a parabolic structure (see the examples of the kinetic equations in Jin (1999), the Euler equations with friction in Chalons, Girardin, and Kokh (2013) or with drag force in Chalons, Massot, and Vié (2015)). In this case, using a standard fractional-step method can give numerical solutions which are bad approximations of the weak solution for the asymptotic system. This is due to consistency errors in the flux discretization that are proportional to $\Delta x/\mu$, where Δx is the spatial discretization step, μ is the relaxation parameter and $\mu \to 0$. In order to avoid an arbitrarily small space step, a modified scheme is proposed in Chalons, Girardin, and Kokh (2013) for example, which consists in accounting for the source terms when computing the numerical fluxes. The scheme is inspired from the well-balanced schemes. The consistency errors are then counterbalanced and better results can be obtained.

In the following, we are using operator splitting methods in two ways:

- to solve the convective part and the source terms of the two-phase systems independently,
- with dimensional splitting, so as to solve separately the spatial components of the numerical flux, for 2D or 3D computations (see more details in Chapter 6).

We will then propose appropriate and high-order methods for the integration of each operator (Chapter 4 and Chapter 5), and the we will verify that the whole strategy presents no spurious splitting error in the second part of Chapter 5.

Resolution of the convective part

A wide range of finite volume methods are dedicated to the resolution of compressible systems of equations, either monophasic or two-phase equations (Godlewski and Raviart (1996); Leveque (2004); Bouchut (2004); Toro (2009) among many others). In Chapter 4, we choose to use Godunov-type schemes (Godunov (1959)), which are based on the resolution (exact or approximate) of Riemann problems at each cell interface, so as to compute the numerical fluxes. One of these methods, a relaxation scheme, is even more used to solve two-phase flow models (Baudin, Berthon, Coquel, Masson, and Tran (2005); Boileau, Chalons, Bourgouin, Terrier, Laurent, De Chaisemartin, and Massot (2010); Pelanti, Bouchut, and Mangeney (2011); Girardin (2014)). We show in Chapter 4 that this scheme may overestimate the CFL constraint at the two-phase interface, leading to very small time steps, which may unnecessarily increase the cost of a numerical computation. Therefore, we propose a new scheme based on the modified scheme from Bouchut (2004), and show the benefits of this scheme, named Relax-B. According to the analysis of Abgrall (1996), we also propose a specific discretization of the equation on α for the 4-equation and 5-equation systems, which ensures that uniform velocity and pressure profiles are preserved.

The issue of the resolution of the compressible two-phase systems in the low-Mach number limit is

addressed in Chapter 6. Two cures are considered:

- as noticed in Grenier, Vila, and Villedieu (2013) and analyzed in Dellacherie (2010), the resolution of these equations on unstructured, or even just triangular grids seem to improve the accuracy of the solution, in certain configurations,
- inspired from the scheme proposed in Chalons, Girardin, and Kokh (2016); Girardin (2014), we propose a modification of Relax-B that makes it accurate in all Mach number regimes.

These cures are tested and discussed in Chapter 9, where we show that using AMR without one of these cures provides dramatically poor solutions.

Source term integrations and ODE solvers

The second step of the operator splitting procedure consists in integrating the source terms operators. For the 4-equation system, this comes down to solve an Ordinary Differential Equation (ODE) in time which reads:

$$\mathrm{d}_t\alpha(t,\mathbf{x}) = \frac{p_1(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x})) - p_2(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x}))}{\epsilon},$$

where density and mass fraction variables are kept constant (since there is no source terms for these variables), and are noted $\overline{\rho}$, $\overline{Y}$ respectively. For the 5-equation system, this integration comes down to solve a system of ODE:

$$\begin{cases} \mathrm{d}_t\alpha &= \dfrac{\overline{\rho}\overline{Y}w}{\sqrt{\nu}} \\ \mathrm{d}_t w &= -\dfrac{\epsilon}{\nu}w \quad + \quad \dfrac{p_1(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x})) - p_2(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x}))}{\overline{\rho}\overline{Y}\sqrt{\nu}} \end{cases}$$

These two ODE are non-linear, because of the term in $\Delta p(\alpha) = p_1(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x})) - p_2(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x}))$, whose expression for linear pressure laws is:

$$\Delta p(\alpha) = p^o + c_1^2\frac{\overline{\rho}\,\overline{Y}}{\alpha} - c_2^2\frac{\overline{\rho}(1-\overline{Y})}{1-\alpha},$$

where p^o is a constant. $\Delta p(\alpha) = 0$ defines the equilibrium state, for which $\alpha = \alpha_{eq}$ and $w = 0$. We can show (see also in Chapter 5) that the solutions of these two ODE tend to this equilibrium state in long times, provided that $\epsilon > 0$. The characteristic time of this relaxation process, τ_{relax}, depends on the values of ϵ and ν, and the smaller ϵ, the shorter τ_{relax}. When τ_{relax} is much smaller than the integration time step Δt, one says that the ODE is **stiff**. In the limit $\tau_{\text{relax}} \ll \Delta t$, the numerical method may become unstable or unable to converge to the equilibrium state (this is the case of explicit solvers). **A-stable** methods are stable whatever the choice of Δt. Furthermore, **L-stable** methods converge towards the equilibrium state in long times.

Numerical resolution of ODE has a long history and much development of solvers dedicated to the resolution of stiff ODE has occurred in the last decades. In particular, implicit multi-step (Hindmarsh (1983); Brown, Byrne, and Hindmarsh (1989a)) or one-step Runge-Kutta (see Hairer and Wanner (1996) and references therein) methods are shown to be stable and can be extended to high orders of accuracy. Moreover, their use can be associated with an adaptive time-stepping strategy, which ensures a bounded numerical error (Hairer, Nørsett, and Wanner (1987); Hairer and Wanner (1996)).

One of these accurate methods, Radau5 (Hairer and Wanner (1996)), has been tested for integrating the 5-equation source term. Due to the non-linearity of the ODE, its solution becomes extremely stiff when α approaches the values of 0 and 1. Thus, in one of the tests, Radau5 was unable to satisfy the required accuracy in a reasonable time step size and the routine stopped with an error message. The details on the results for the problematic case are given in Appendix B. Consequently, in order to get an approximate solution to the ODE, we must either better understand the dynamics of the ODE and the cases when it becomes stiff, that may help choosing among the existing solvers which is the best adapted. Or one can design an approximate solver, whose accuracy is maybe lower than already existing methods, but

which ensures the stability of the solution. In Chapter 5, we consider the second option, and propose some approximate solvers for the integration of the 4-equation and 5-equation source terms. These approximate solvers present also the advantage of a low computational cost comparing with the high-order solvers. For a first study, these approximate solvers may be interesting to consider. Nevertheless, because of the AMR strategy, that is discussed in the next paragraph, the cost of the ODE integration is not really a limiting factor: it is indeed local to each cell of the mesh and easily distributed among the processes. We could then afford higher order resolutions in future works.

Improving the resolution of the two-phase interface

The last aspect which is addressed in Chapter 6 concerns the reduction of the numerical diffusion in cases of separated two-phase flow simulations. Let us recall that in the interface capturing approach, the two-phase interface is not reconstructed and during a computation, it can be spread due to numerical diffusion. We discuss here a series of methods that permit to improve the accuracy of the numerical solution.

Our choice is to use AMR, in order to increase the space resolution near interfacial zones, while solving the monophasic zones at a reasonable cost. AMR techniques are indeed efficient tools for configurations involving localized fronts (see the discussion in Duarte (2012)). Now, to ensure an accuracy of the numerical solution which is not damaged by the adapted grid, *i.e.* a compression error negligible with respect to the other numerical errors, the AMR methods necessitate to design refinement criteria and projection formulae to set the values of the solution in cells that are refined or coarsened. Refinement strategies are proposed in Chapter 6.

A consequence of the use of AMR methods, and in particular in the case of cell-based AMR (see the definition in the introduction of Part III), is that the solution has now to be computed on **non-conforming** grids: the shape of the cells is still regular, but across their faces, they can have more than one neighbor. In Chapter 6, we propose to adapt numerical methods originally used on Cartesian grids for these non-conforming grids and to benefit from the *regularity* of the grid to use dimensional-splitting or 1D slope reconstructions in a second-order MUSCL method.

Remark 15. *For unstructured meshes, the monoslope method is one of the standard approach. It is based on the computation of a unique gradient within each cell. Values at the cell interfaces are then interpolated with this gradient. An approach where each face of the cell has its own local gradient was developed by Stoufflet, Periaux, Fezoui, and Dervieux (1987); Buffard and Clain (2010) and is called multisplope. It is shown to be more accurate than the monoslope approach (Murrone and Guillard (2008)), but could be applied only on triangular (2D) and tetrahedral (3D) grids. Recent developments have improved the multislope approach (see Le Touze (2015); Dupif, Lagarde, Boileau, Laurent, and Massot (2015); Dupif (2017); Dupif, Dupays, Larat, and Massot (2017)).*

In our discretization strategy, we are indeed interested in using second-order methods. Many methods can be used to get second-order (or higher order) finite-volume schemes, among which:

- **MUSCL** scheme: this method aims at getting a second-order in space solution, using linear reconstruction of the solution. An extension to 3^{rd} order has also been developed using parabolic reconstructions (Kermani, Gerber, and Stockie (2003)),
- **ENO/WENO** schemes: have been developed and are largely used to get high-order schemes (more than second-order in space). They are based on the interpolation of different degrees polynomials, using a predefined cell stencil (ENO), that might be weighted (WENO), so as to avoid oscillations of the solution as much as possible (Jiang and Shu (1996)),
- **DG** method: the Discontinuous Galerkin method (Cockburn and Shu (1989); Cockburn and Shu (1998b); Cockburn and Shu (1998a)) is issued from the finite element techniques, but uses in fact a combination of finite elements and finite volumes. It indeed uses a subdivision of each cell in quadrature points where a trial space of functions is used to solve the equations, but contrary to classic finite elements, these functions are not continuous across the cells interfaces. There, fluxes are computed as in classic finite volume methods and are used to update the values at the quadrature points in each neighboring cell.

- **Runge-Kutta** integration: the methods previously presented are high-order schemes in terms of spatial resolution. Time integration can also be improved, by considering techniques of integration for ODEs. Among the mostly used methods is the Runge-Kutta integrations, which can give schemes with orders of accuracy higher than 2. It consists in computing the different terms, related to different sub-steps integrations, of a specified averaging formula. For example, the second order integration from t^n to $t^{n+1} = t^n + \Delta t$ writes:

$$\begin{aligned} \mathbf{W}^* &= \mathbf{W}^n + \frac{\Delta t}{2}\mathcal{R}\left(\mathbf{W}^n\right), \\ \mathbf{W}^{n+1} &= \mathbf{W}^n + \Delta t\mathcal{R}\left(\mathbf{W}^*\right), \end{aligned}$$

  where $\mathcal{R}$ is the discrete value of the flux divergence for systems such as (6.1).

- **MUSCL-Hancock** scheme is a second order in space and time method, based on the MUSCL linear reconstruction and a prediction of the local solution at an intermediate step. It is detailed in Chapter 6,

- when using MUSCL methods, the second order in space and in time involves multiple integration steps and, at the end of the process, a large stencil of neighboring data is involved. These methods generally show to be Total Variation Diminishing (TVD), which is a property we define below.

Without being unstable, some numerical schemes can develop oscillations near strong discontinuities of the solution. A way in characterizing the amount of these oscillations is by looking at the total variation (TV) of the solution. For a scalar function q, it can be defined by (see Leveque (2004) for example):

$$TV(q) = \int_{-\infty}^{+\infty} \left|q'(x)\right| dx.$$

If q is differentiable, then q' is the derivative. If q is piecewise constant (on a given grid), then the total variation can be written as:

$$TV(q) = \sum_{i=-\infty}^{+\infty} \left|q(x_i) - q(x_{i-1})\right|.$$

If oscillations occur, then the total variation of the solution increases. That is why we are looking for a non-increasing property for numerical methods, that is in fact called Total Variation Diminishing (Harten (1983)), and that writes, comparing the solution from time t^n to time t^{n+1}:

$$TV\left(q(t^{n+1})\right) \leqslant TV\left(q(t^n)\right). \tag{4.27}$$

Remark 16. *The notion of total variation is well-defined for scalar equations, but its extension to systems of equations is far from being obvious. The TVD property could be extended for a scheme if it is TVD for scalar laws. Or one could define a norm $|\cdot|$ for the vector of variables, like a L1-norm using the projection on the eigenvectors of the system for example (as in Godlewski and Raviart (1996)).*

The TVD property has lead to the design of *slope limiters* in the MUSCL reconstruction method (Haider (2009)). However, these limiters present the drawback of clipping the extrema of the solution, and the order of accuracy of the method may be lost in these extremal zones. Another approach is proposed with the One-Step-Monocity-Preserving (OSMP) methods (Daru and Tenaud (2004)): these methods provide a high-order resolution using only one layer of neighboring cells and they allow a better resolution near the extrema.

Summary and organization of the following chapters

Let us sum up the features of the global strategy that will be developed in the following:

- an operator splitting method is used, that enables to integrate independently the convective part of the equations and their source terms, and that is also used for dimensional splitting in 2D and 3D configurations,

- relaxation approximate Riemann solvers to solve the convective part of the equations and an appropriate discretization of the equation on α,
- an ODE solver for the integration of the source terms that is able to handle the stiffness of the source terms,
- high-order methods that help to reduce the numerical diffusion in the two-phase interfacial zones,
- application of these methods in an AMR context, where the solution must be computed on non-conforming grids.

Although the frame of our study is the resolution of barotropic homogeneous systems of equations, the methods that are presented in the three following chapters should apply to more general two-phase equations, with some adaptations.

Schemes for the resolution of the convective part are presented in Chapter 4. We will take care of the discretization of the equation on the volume fraction α. We will also consider relaxation methods and propose a new scheme, which is more convenient than the standard relaxation approach for the resolution of two-phase flow configurations. The integration of the source terms will be addressed in Chapter 5. Indeed, due to the strong stiffness of the source terms of the 4-equation and 5-equation models, a dedicated strategy is necessary. In Chapter 5, we will also test the whole discretization strategy (operator splitting, approximate Riemann solver and source term solver) for tests of the propagation of acoustic waves. The objective will be to recover numerically the dispersion relations of Chapter 2, and the asymptotic behaviors of the different models. Chapter 6 will address the adaptation of standard finite volume methods on non-conforming grids. We will also investigate the low-Mach problem in Chapter 6, and consider using a simple correction to the usual Godunov-type schemes, as proposed in Chalons, Girardin, and Kokh (2016).

Chapter 4

Approximate and relaxation Riemann solvers for the resolution of two-phase equations

The three two-phase models that must be discretized will be considered in the following generic form:

$$\left\{\begin{array}{rcrcl} \partial_t\left(\rho\mathbf{u}\right) & + & \mathrm{div}\left(\rho\mathbf{u}\mathbf{u}^T + P\,\mathbb{Id}\right) & = & 0 \\ \partial_t\rho & + & \mathrm{div}\left(\rho\mathbf{u}\right) & = & 0 \\ \partial_t\mathbf{V} & + & \mathbf{u}^T\,\boldsymbol{\nabla}\mathbf{V} & = & \mathbf{S}(\rho,\mathbf{V}) \end{array}\right. \tag{4.1}$$

where $\mathbf{V} = Y$ for the 3-equation system, $\mathbf{V} = (Y,\alpha)^T$ for the 4-equation system and $\mathbf{V} = (Y,\alpha,w)^T$ for the 5-equation system. This chapter is devoted to the resolution of the left-hand side of the equations, namely the convective part. We are looking for robust methods, which are also accurate for a *reasonable* complexity. We are interested in standard finite volume schemes for compressible equations (Leveque (2004)). These methods are briefly presented in the first part of the present chapter. Then, the numerical fluxes are computed by approximate Riemann solvers (Toro (2009); Bouchut (2004)) and more specifically by relaxation solvers (Jin and Xin (1995); Bouchut (2004)). We will show a shortcoming of Suliciu's relaxation scheme (Suliciu (1990); Bouchut (2004)) when two-phase systems are solved, and we will propose a modification to this method, inspired by Bouchut (2004). This modification is a first proposal, and needs further developments and studies for the cases of the two-phase systems. This is why, we also present a HLLC method, which has shown good results in practice. Another delicate point concerns the discretization of some of the transport equations (equations for $\mathbf{V}$ variable). Indeed, from the study of Abgrall (1996); Saurel and Abgrall (1999), one knows that an inappropriate scheme for the equation on α can generate pressure oscillations. This is why we also propose a discretization, following the lines of Godlewski and Raviart (1996); Kokh and Lagoutière (2010) for this variable. The chapter ends with some numerical tests that verify the proper functioning of the numerical scheme, despite some remaining issues.

4.1 Overview of the finite volume method

4.1.1 Space and time discretization

In the following, we are considering systems that are solved in periodic or infinite spatial domains, so as to avoid the issues related to the boundary treatments. The treatment of boundary conditions is indeed a complex question and goes beyond the scope of this chapter. Thus, let us consider an initial value problem that writes:

$$\left\{\begin{array}{rcrcll} \partial_t\mathbf{W}(t,\mathbf{x}) & + & \mathrm{div}_{\mathbf{x}}\left(\mathbf{F}\left(\mathbf{W}(t,\mathbf{x})\right)\right) & = & 0, \quad t>0, \quad \mathbf{x}\in\mathbb{R} \\ & & \mathbf{W}(t=0,\mathbf{x}) & = & \mathbf{W}^0(\mathbf{x}) \end{array}\right. \tag{4.2}$$

where $\mathbf{W} : \mathbb{R}^+ \times \mathbb{R}^3 \mapsto \mathbb{R}^N$ is a vector of N conserved variables and $\mathbf{F} \in \mathcal{C}^2\left(\mathbb{R}^N; \mathbb{R}^N\right)$ is their physical flux. This formulation may correspond to a conservative formulation of equation (4.1), without the source terms (the source terms will be integrated in a second step, according to the operator splitting procedure).

In most cases, analytical solutions for the problem (4.2) are out of reach. Then we must resort to approximate solutions, made of a discrete set of values in space over a discrete series of time steps. Let us note $\mathbf{W}_i^n$ such an approximate solution, given near a point $\mathbf{x}_i$ and at time t^n.

Let us choose a partition of the space, that remains fixed over time, defining a spatial mesh. We consider then a new problem, made of an approximation of $\mathbf{W}^0$ and evolution laws for the values $\mathbf{W}_i^n$. One of the simplest approximations of $\mathbf{W}^0$ is a piecewise constant function $\mathbf{W}_i^0$, defined as the volume average of $\mathbf{W}^0$ over each cell $\mathcal{V}_i$ of the mesh such that $\mathbf{x}_i \in \mathcal{V}_i$:

$$\mathbf{W}_i^0 = |\mathcal{V}_i|^{-1} \int_{\mathcal{V}_i} \mathbf{W}^0(\mathbf{x}) \mathrm{d}V, \tag{4.3}$$

where $|\mathcal{V}_i|$ is the volume of cell $\mathcal{V}_i$. This piecewise constant approximate solution is illustrated in Figure 4.1.

Let us integrate the evolution law (4.2) from time $t^n > 0$ to time $t^{n+1} > t^n$ and over the cell $\mathcal{V}_i$:

$$\begin{aligned} 0 &= \int_{t^n}^{t^{n+1}} \int_{\mathcal{V}_i} \partial_t \mathbf{W}(t, \mathbf{x}) \mathrm{d}V \, \mathrm{d}t + \int_{t^n}^{t^{n+1}} \int_{\mathcal{V}_i} \mathrm{div}_{\mathbf{X}} \left(\mathbf{F}\left(\mathbf{W}(t, \mathbf{x})\right)\right) \mathrm{d}V \, \mathrm{d}t \\ &= |\mathcal{V}_i| \left(\mathbf{W}_i^{n+1} - \mathbf{W}_i^n\right) + \int_{t^n}^{t^{n+1}} \int_{\partial \mathcal{V}_i} \mathbf{F}\left(\mathbf{W}(t, \mathbf{x})\right) \cdot \mathbf{n}_i \, \mathrm{d}\Sigma \, \mathrm{d}t \end{aligned}$$

where $\partial \mathcal{V}_i$ denotes the border of $\mathcal{V}_i$, measured by $\mathrm{d}\Sigma$ and $\mathbf{n}_i$ is the outgoing normal at the interface.

Now, let us assume that the cells $\mathcal{V}_i$ are polyhedral and let us denote $\Gamma_{i,j}$ one face of cell $\mathcal{V}_i$, $\mathbf{n}_{i,j}$ its outgoing normal vector. The neighboring cell at that face will be noted $\mathcal{V}_j$. Then we have:

$$\partial \mathcal{V}_i = \sum_j \Gamma_{i,j}.$$

An approximation of $\int_{t^n}^{t^{n+1}} \int_{\partial \mathcal{V}_i} \mathbf{F}\left(\mathbf{W}(t, \mathbf{x})\right) \cdot \mathbf{n}_i \, \mathrm{d}\Sigma \, \mathrm{d}t$ may assume a uniform numerical flux $\mathbf{F}_{i,j}$ over each face and constant throughout the time step $\Delta t = t^{n+1} - t^n$, such that:

$$\Delta t \, |\Gamma_{i,j}| \, \mathbf{F}_{i,j} \approx \int_{t^n}^{t^{n+1}} \int_{\Gamma_{i,j}} \mathbf{F}\left(\mathbf{W}(t, \mathbf{x})\right) \cdot \mathbf{n}_{i,j} \mathrm{d}\Sigma \, \mathrm{d}t \,,$$

with $|\Gamma_{i,j}|$ the length (in 2D) or surface (in 3D) of the face. The approximate fluxes $\mathbf{F}_{i,j}$ may be given as functions of the approximate solution at time t^n: the scheme is then said to be **explicit** in time; or as functions of the approximate solution at time t^{n+1}: the scheme is then **implicit** in time. For both options, the scheme accuracy will be at most first-order in time (we are defining the order of accuracy of the numerical solutions in section 4.1.2). Moreover, the choice of piecewise constant solutions per cell leads to a scheme, which is at most first-order accurate in space. Within this approach, higher-order methods will be presented in Chapter 6.

Another approach to finite volume schemes, which can also be considered as a finite difference discretization, is represented by the schemes based on a Lax-Wendroff flux formulation (Lax and Wendroff (1960); Hirsch (1990)). Lax-Wendroff scheme is second-order accurate and more accurate time integration can be considered with the One-Step Monotonicity Preserving approach of Daru and Tenaud (2004). These schemes are based on the Lax-Wendroff scheme and on recursive modifications of the flux function so as to reduce the consistency errors. Although these methods present many advantages, they are not further considered here: we choose to use Godunov-type methods in the following (Godunov (2008)).

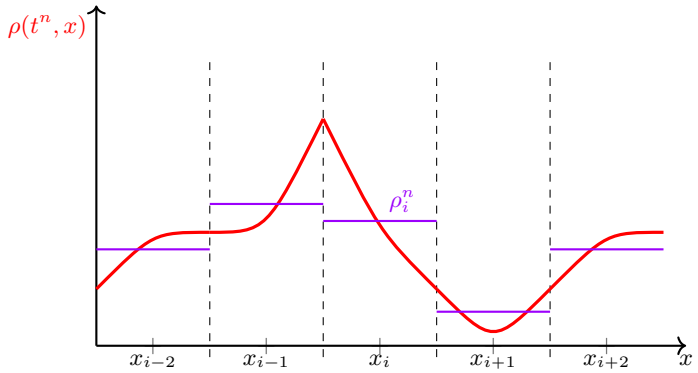


Figure 4.1: *Volume averaged and piecewise constant approximation (in purple) of the continuous solution (in red).*

Finally, the finite volume scheme reads, for a polyhedral cell $\mathcal{V}_i$:

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n + \frac{\Delta t}{|\mathcal{V}_i|} \sum_j |\Gamma_{i,j}| \mathbf{F}_{i,j} \,. \tag{4.4}$$

The discretization using finite volume schemes is intrinsically conservative, provided that the fluxes at the interface are uniquely defined for both neighboring cells. Since we consider only periodic or infinite boundary conditions, this means that for any time $t^n > 0$:

$$\sum_i \mathbf{W}_i^n |\mathcal{V}_i| = \int_{\mathbb{R}} \mathbf{W}^0(\mathbf{x}) d\mathbf{x}.$$

Now, we would like to compare the entropic solution $\mathbf{W}$ of the continuous problem (4.2) with the approximate solution $(\mathbf{W}_i^n)$ obtained with the discrete initial condition (4.3) and the scheme (4.4). More particularly, we expect $(\mathbf{W}_i^n)$ to *converge* toward $\mathbf{W}$ when the space and time steps tend to zero. Some information on this convergence is brought by the properties of the numerical flux $\mathbf{F}_{i,j}$ which are discussed in the next paragraph. In the following, we only consider the explicit time integration and assume that $\mathbf{F}_{i,j}$ is given by:

$$\mathbf{F}_{i,j} = \mathcal{F}\left(\mathbf{W}_i^n, \mathbf{W}_j^n\right), \tag{4.5}$$

where $\mathcal{F}$ is the *numerical flux* and is Lipschitz with respect to $\mathbf{W}_i^n$ and $\mathbf{W}_j^n$.

4.1.2 Properties of the scheme and of the numerical fluxes

Convergence

For scalar equations ($N = 1$), it is possible to prove the convergence of the discrete solution towards the unique entropic solution of (4.2), under the assumptions that the numerical flux $\mathcal{F}$ is consistent with the continuous flux $\mathbf{F}$ and that the scheme is stable (see Godlewski and Raviart (1996) for example).

On the other hand, for systems of equations ($N > 1$), studying the convergence of the numerical schemes is still an open issue. However, by analogy with the scalar case, some properties of the schemes help the discrete solution to converge towards the entropic weak solution of system (4.2). These properties are the **consistency** of the numerical flux, the **preservation of invariant domains** and the existence of a **discrete entropy inequality** (the last two properties help ensuring the **stability** of the scheme):

1. The consistency of the numerical flux simply means that, when the numerical solution $\mathbf{W}_i^n$ converges towards the continuous solution, then the numerical flux $\mathcal{F}$ converges towards the physical flux $\mathbf{F}$. In other words, one must have $\mathcal{F}(\mathbf{W}, \mathbf{W}) = \mathbf{F}(\mathbf{W})$.
2. An invariant domain $\Omega \in \mathbb{R}^N$ is such that $\mathbf{W}(t, \mathbf{x}) \in \Omega$ for any $(t, \mathbf{x})$. For instance for gas dynamics equations, this domain is made of all solutions with positive density and positive internal energy. The scheme preserves invariant domains if $\mathbf{W}_i^n \in \Omega$, $\forall i$, $\forall n$.

3. For an entropy-entropy flux pair $(\eta, \mathbf{G})$ for system (4.2), the numerical scheme satisfies a discrete entropy inequality if, for a discrete entropy flux function $\mathcal{G}$ consistent with $\mathbf{G}$, one has:

$$\eta\left(\mathbf{W}_i^{n+1}\right) - \eta\left(\mathbf{W}_i^n\right) + \frac{\Delta t}{|\mathcal{V}_i|}\sum_j |\Gamma_{i,j}|\mathcal{G}\left(\mathbf{W}_i^n, \mathbf{W}_j^n\right) \leqslant 0. \tag{4.6}$$

In general cases, properties 2. and 3. necessitate that a CFL (for Courant-Friedrichs-Lewy) condition is satisfied: this condition gives an upper bound to the value of the time step Δt according to the values of the system characteristic velocities (the absolute value of the eigenvalues) and a characteristic length of the spatial discretization.

For the finite volume scheme defined by (4.4), properties 2. and 3. just above are equivalent to the following properties on the numerical flux when a CFL condition is satisfied (see Bouchut (2004)):

- the numerical flux preserves invariant domains, in the sense of Bouchut (2004), p. 15,
- the numerical flux is entropic, in the sense of Bouchut (2004), page 17.

Now that we have these rules that help design numerical fluxes, we can wonder how accurate this scheme is. To answer this question, an analysis of the truncation errors can be performed.

Equivalent continuous equations and truncation errors

In the case of smooth solutions, for each discrete scheme, such as (4.4), one can recover an **equivalent continuous equation** (also called **modified equation**) by letting Δt and the spatial discretization tend to zero. Let Δx be a characteristic length of the spatial discretization (for instance $\Delta x = \max_{i,j}\left(\frac{|\mathcal{V}|_i}{|\Gamma_{i,j}|}\right)$), then the equivalent equation may have the form (see Leveque (1992) for example):

$$\partial_t \mathbf{W}(t,\mathbf{x}) + \mathrm{div}_{\mathbf{x}}\left(\mathbf{F}\left(\mathbf{W}(t,\mathbf{x})\right)\right) = O\left(\Delta t^{\alpha}\right) + O\left(\Delta x^{\beta}\right). \tag{4.7}$$

which is close to the physical continuous equation from (4.2), up to the **truncation error terms**: $O\left(\Delta t^{\alpha}\right)$ and $O\left(\Delta x^{\beta}\right)$. The exponents α and β must be positive, and they can be shown to define the order of convergence of the scheme in time and space respectively, when the scheme is stable enough. Then, when one talks about schemes that are first-order in time, one assumes that $\alpha = 1$.

In the case of discontinuous solutions (like shocks or contact discontinuities) we refer to Leveque (1992), where the analysis consists in considering approximate continuous solutions. It is shown that the scheme order of accuracy can be decreased in discontinuous cases compared to smooth solutions.

4.1.3 Godunov-type schemes

Godunov exact solver

With Godunov's approach, the approximate fluxes $\mathbf{F}_{i,j}$ are defined using the exact solution of the Riemann problem at the cells interfaces $\mathcal{W}^{\mathcal{R}}\left(\mathbf{W}_i^n, \mathbf{W}_j^n, \xi\right)$, with $\xi = t/x$. Riemann problems are defined by an initial condition made of two constant states. For this problem, the weak entropic solution of hyperbolic equations like (4.2) may be determined exactly, using the nature of the waves associated with the eigenvalues of the Jacobian of (4.2), Rankine-Hugoniot jump relations across the shock waves and contact discontinuities, and Riemann invariants through the rarefaction waves. Then, Godunov's numerical flux reads:

$$\mathbf{F}_{i,j} = \mathbf{F}(\mathbf{W}_i^n) - \int_{\xi=-\infty}^{0}\left(\mathcal{W}^{\mathcal{R}}\left(\mathbf{W}_i^n, \mathbf{W}_j^n, \xi\right) - \mathbf{W}_i^n\right) d\xi.$$

Since the exact solution of the Riemann problem is consistent with the weak continuous solution of (4.2), it pertains to the convex invariant domains and is entropic, one can show that Godunov's scheme satisfies the three properties of flux consistency, preservation of the invariant domains and discrete entropy inequality cited above, under the following **Courant-Friedrichs-Lewy** constraint on the time

step:

$$\Delta t \leqslant \frac{1}{2} \min_{i,j} \frac{\Delta x_i}{|\sigma_{i,j}^{\pm}|}, \tag{4.8}$$

where $|\sigma_{i,j}^{\pm}|$ is the maximum information velocity inside the Riemann problem at face $\Gamma_{i,j}$.

Computing the exact solution of the Riemann problem can be quite cumbersome and sometimes computationally expensive. That is why approximate Riemann solvers may be used to design the numerical fluxes.

Approximate Riemann solvers

It is possible to design approximate Riemann solvers $\mathcal{W}^*\left(\mathbf{W}_i^n, \mathbf{W}_j^n, \xi\right)$ by mimicking the properties of the Riemann problem exact solution. Namely, we require that:

- this solver is consistent:

$$\mathcal{W}^*\left(\mathbf{W}, \mathbf{W}, \xi\right) = \mathbf{W},$$

- this solver preserves invariant domains:

$$\text{if} \quad \mathbf{W}_i^n, \mathbf{W}_j^n \in \Omega \quad \text{then} \quad \mathcal{W}^*\left(\mathbf{W}_i^n, \mathbf{W}_j^n, \xi\right) \in \Omega,$$

- the flux associated with this solver is entropic.

For instance, simple solvers consider approximate solutions made of constant states separated by discontinuities. The resolution is done by using jump relations through these discontinuities. Examples of such solvers are the HLL and HLLC schemes Toro (2009); Bouchut (2004).

4.2 Proposition of a numerical scheme adapted to the two-phase barotropic systems of equations

Now that we have introduced some useful notions, let us go back to the two-phase systems of equations (4.1) and look for a good numerical scheme to solve them. First, we consider a general formulation of the numerical scheme, that can be applied to any of the three two-phase models.

In this part, we consider the following 1D system of conservation laws (1D formulation of system (4.1)):

$$\left\{\begin{array}{rcrcl} \partial_t\left(\rho u\right) & + & \partial_x\left(\rho u^2 + P\right) & = & 0, \\ \partial_t \rho & + & \partial_x\left(\rho u\right) & = & 0, \\ \partial_t \mathbf{V} & + & u\partial_x \mathbf{V} & = & 0, \end{array}\right. \tag{4.9}$$

where $\mathbf{V}$ contains the other components of the velocity field: u_y and u_z ; the mass fraction Y, the volume fraction α for the 4- and 5-equation systems, and w for the 5-equation system. Let us assume that the total number of equations is N.

The nature of the waves associated with the eigenvalues of (4.9) have been introduced in the introduction of Part II, and the exact solutions to the Riemann problems are presented in Appendix A. However, these solutions may be complex to compute and consequently a Godunov scheme would probably be computationally expensive. This is why we are interested in approximate Riemann solvers and more specifically in relaxation methods.

4.2.1 Suliciu's relaxation method

In non-linear systems of equations, the non-linearities associated with the flux function can be at the origin of numerical difficulties. The idea of relaxation methods is to modify these systems of equations,

so as to bypass their non-linearities. Despite the fact that the modifications lead to an enlarged system, its resolution is easier.

This idea comes from the works of Jin and Xin (1995), where the resulting system is fully linear, and has been adapted and studied in many works. Among the variants of the relaxation methods, one finds Suliciu's approach (Suliciu (1990); Bouchut (2004)), which is well-suited for Euler-type equations, and which consists in linearizing only the pressure term from the momentum equation (and not the whole momentum flux as in Jin and Xin (1995)). This method has also been used and studied in Liu (1987); Chen, Levermore, and Liu (1994); Chalons and Coquel (2005); Chalons and Coulombel (2008); Boileau, Chalons, and Massot (2015).

Suliciu's relaxation method consists in considering a new variable Π such that Π tends to $P(\rho, \mathbf{V})$ with a given rate of convergence ϵ. In practice in relaxation methods, this parameter is considered as infinitely small. The equation on the new variable Π is chosen so as to mimic the equation on the real pressure P. For system (4.9), the equation on P can be deduced from the equations on ρ and $\mathbf{V}$. It writes:

$$\partial_t(\rho P) + \partial_x(\rho P u) + \rho^2 c^2 \partial_x u = 0. \tag{4.10}$$

Then, let us assume that the equation on Π writes:

$$\partial_t(\rho\Pi) + \partial_x(\rho\Pi u) + a^2 \partial_x u = \rho \frac{P - \Pi}{\epsilon}, \tag{4.11}$$

where a is a constant that has the dimension of a momentum. An accurate definition of a will be discussed just below. The new system of $N + 1$ equations, with the relaxation effects on the pressure writes in its conservative form:

$$\left\{ \begin{array}{rcrcl} \partial_t(\rho u) & + & \partial_x\left(\rho u^2 + \Pi\right) & = & 0 \\ \partial_t(\rho\Pi) & + & \partial_x\left(\rho\Pi u + a^2 u\right) & = & \rho\frac{P-\Pi}{\epsilon} \\ \partial_t\rho & + & \partial_x(\rho u) & = & 0 \\ \partial_t(\rho\mathbf{V}) & + & \partial_x(\rho\mathbf{V}u) & = & 0 \end{array} \right. . \tag{4.12}$$

The new numerical strategy for system (4.9) consists in solving the convective part of system (4.12), namely:

$$\left\{ \begin{array}{rcrcl} \partial_t(\rho u) & + & \partial_x\left(\rho u^2 + \Pi\right) & = & 0 \\ \partial_t(\rho\Pi) & + & \partial_x\left(\rho\Pi u + a^2 u\right) & = & 0 \\ \partial_t\rho & + & \partial_x(\rho u) & = & 0 \\ \partial_t(\rho\mathbf{V}) & + & \partial_x(\rho\mathbf{V}u) & = & 0 \end{array} \right. , \tag{4.13}$$

and then integrating the source term in the limit $\epsilon = 0$. More precisely, this second step comes down to updating the value of Π so as to get $\Pi = P(\rho, \mathbf{V})$.

The advantage of this strategy is that it involves simple computations, as we will see through the exact resolution of the Riemann problem, and under a **subcharacteristic condition**, that the global scheme is entropic.

Subcharacteristic condition and entropy:

According to Bouchut (2004), if the constant a satisfies the following **subcharacteristic condition**:

$$a > \rho c \tag{4.14}$$

for all values of ρ and c in the Riemann problem, then there exists an entropy η^ϵ and an entropy flux $\mathbf{G}^\epsilon$ for the relaxation scheme, whose expressions are given in Bouchut (2004). Then, in Bouchut (2004), it is shown that, if the entropy inequality is satisfied for η^ϵ, then it is also satisfied for η, which is the entropy of the original system (4.9).

Another way to show the importance of this subcharacteristic condition is to perform a Chapman-Enskog expansion on the variable Π as in Chen, Levermore, and Liu (1994) and use its first order variations to recover the system diffusive character. Indeed, let us consider that:

$$\Pi = \Pi^{(0)} + \epsilon\Pi^{(1)} + O\left(\epsilon^2\right).$$

Then, multiplying equation (4.11) by ϵ on both sides and letting ϵ tend to 0 gives at zeroth order:

$$\Pi^{(0)} = P.$$

Equation (4.11) now writes:

$$\partial_t \Pi + u\partial_x \Pi + \frac{a^2}{\rho}\partial_x u = -\Pi^{(1)}.$$

At zeroth order in ϵ one gets:

$$\partial_t P + u\partial_x P + \frac{a^2}{\rho}\partial_x u = -\Pi^{(1)}.$$

Moreover, since P verifies equation (4.10), we have:

$$-\Pi^{(1)} = \left(\frac{a^2}{\rho} - \rho c^2\right)\partial_x u. \tag{4.15}$$

Let us consider the momentum equation and note that, by keeping the first order term in ϵ, it writes:

$$\begin{aligned}\partial_t(\rho u) + \partial_x\left(\rho u^2 + P\right) &= -\epsilon\partial_x \Pi^{(1)} + O(\epsilon^2)\\ &= \epsilon\partial_x\left(\left(\frac{a^2}{\rho} - \rho c^2\right)\partial_x u\right) + O(\epsilon^2).\end{aligned}$$

The right-hand side term of the last line is dissipative, with a viscosity coefficient given by $\frac{a^2}{\rho} - \rho c^2$. Hence, the modified equation is well-posed provided the viscosity is positive: this condition is equivalent to the relation (4.14).

Relation (4.14) is also called Whitham's subcharacteristic condition (see Whitham (1974); Liu (1987); Jin and Xin (1995)). In the framework of a local Riemann problem with two states $\mathbf{W}_L$ and $\mathbf{W}_R$, we can define a local constant a by

$$a \;=\; \max\left(\rho_L c_L, \rho_R c_R\right). \tag{4.16}$$

Riemann problem resolution:

Let us now design a numerical scheme for system (4.13). One can easily show that this system is hyperbolic. It has three eigenvalues:

$$\lambda_1 = u - a/\rho\,, \quad \lambda_2 = u + a/\rho \quad \text{and} \quad \lambda_{2+i} = u, \quad i = 1, ..., N-1.$$

In the basis $\mathbf{U} = \left(u, \Pi, \rho, \mathbf{V}^T\right)^T$, the associated eigenvectors are:

$$\mathbf{P}_1 = \left(1, -a, -a^2/\rho, \mathbf{0}\right)^T, \quad \mathbf{P}_2 = \left(1, a, a^2/\rho, \mathbf{0}\right)^T, \quad \mathbf{P}_{2+i} = \left(0, 0, \left(\frac{\delta_{i=j}}{\gamma_j}\right)^T_{j\leqslant N+1}\right)^T.$$

with $\delta_{i=j} = 1$ if $j = i$, $\delta_{i=j} = 0$ otherwise. These vectors are obviously linearly independent. Moreover, the associated fields are linearly degenerate: we can easily verify that $\boldsymbol{\nabla}_{\mathbf{U}}\lambda_j \cdot \mathbf{P}_j = 0$, $j \in [\![1, N+1]\!]$. The waves are then all contact discontinuities and the exact Riemann problem can be solved exactly for the relaxation system (4.13) by using the jump relations across these waves.

The Riemann problem is composed of four states, separated by the three contact discontinuities (see Figure 4.2b). Let us denote these states by $\mathbf{U}_L$, $\mathbf{U}_L^*$, $\mathbf{U}_R^*$ and $\mathbf{U}_R$. Let us also note that the eigenvalues are Riemann invariants across the waves they are associated with. We have thus a first set of relations:

$$u_L - \frac{a}{\rho_L} = u_L^* - \frac{a}{\rho_L^*}, \tag{4.17}$$

$$u_L^* = u_R^*, \tag{4.18}$$

$$u_R + \frac{a}{\rho_R} = u_R^* + \frac{a}{\rho_R^*}. \tag{4.19}$$

According to (4.18), we will consider that $u^* = u_L^* = u_R^*$ (see Figure 4.2). Let us now look at the jump relations across the wave related with u. There we have, for the momentum equation:

$$u^* \left(\rho_R^* u^* - \rho_L^* u^*\right) = \rho_R^* {u^*}^2 + \Pi_R^* - \rho_L^* {u^*}^2 - \Pi_L^* \tag{4.20}$$

and we can easily deduce that $\Pi_L^* = \Pi_R^* = \Pi^*$. Let us introduce an auxiliary variable j such that $j = u_L - \frac{a}{\rho_L} = u^* - \frac{a}{\rho_L^*}$. The Rankine-Hugoniot jump relation for the momentum equation across the wave associated with j writes:

$$\begin{aligned} j\left[\rho_L^* u^* - \rho_L u_L\right] &= \rho_L^* {u^*}^2 + \Pi^* - \rho_L u_L^2 - \Pi_L, \\ &\Leftrightarrow \\ u^*\left(\rho_L^* j - \rho_L^* u^*\right) - u_L\left(\rho_L j - \rho_L u_L\right) &= \Pi^* - \Pi_L, \\ &\Leftrightarrow \\ -a u^* + a u_L &= \Pi^* - \Pi_L. \end{aligned}$$

A similar development across the wave associated with the eigenvalue $\lambda_3 = u + \frac{a}{\rho}$ gives:

$$a u_R - a u^* = \Pi_R - \Pi^*$$

and from the last two equations, it is now easy to deduce formulae for u^* and Π^*. Finally, let us look at $\mathbf{V}$, and the jump relation across the first wave:

$$\begin{aligned} j\left[\rho_L^* \mathbf{V}_L^* - \rho_L \mathbf{V}_L\right] &= \rho_L^* \mathbf{V}_L^* u^* - \rho_L \mathbf{V}_L u_L \\ &\Leftrightarrow \\ \mathbf{V}_L^*\left(\rho_L^* j - \rho_L^* u^*\right) - \mathbf{V}_L\left(\rho_L j - \rho_L u_L\right) &= 0 \\ &\Leftrightarrow \\ -a\mathbf{V}_L^* + a\mathbf{V}_L &= 0 \\ &\Leftrightarrow \\ \mathbf{V}_L^* &= \mathbf{V}_L \end{aligned}$$

The same reasoning leads to $\mathbf{V}_R^* = \mathbf{V}_R$.

To conclude, the two intermediate states are defined by:

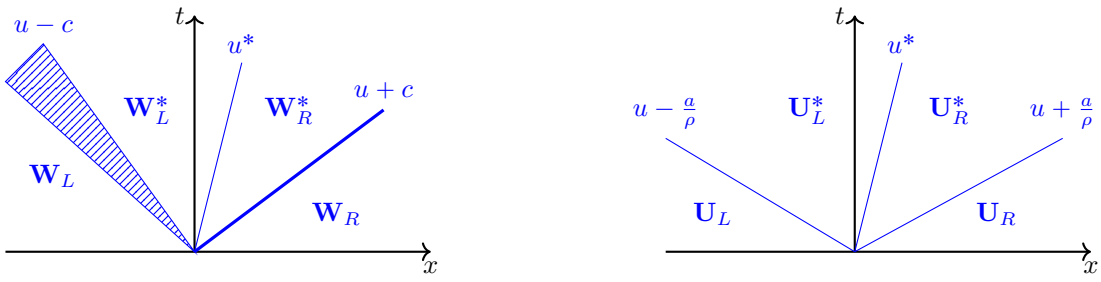
$$\begin{cases} u^* &= \dfrac{u_L + u_R}{2} + \dfrac{\Pi_L - \Pi_R}{2a} \\ \Pi^* &= \dfrac{\Pi_L + \Pi_R}{2} + a\dfrac{u_L - u_R}{2} \\ \rho_L^* &= \left(\dfrac{1}{\rho_L} + \dfrac{u^* - u_L}{a}\right)^{-1} \\ \mathbf{V}_L^* &= \mathbf{V}_L \end{cases} \quad \text{and} \quad \begin{cases} u^* &= \dfrac{u_L + u_R}{2} + \dfrac{\Pi_L - \Pi_R}{2a} \\ \Pi^* &= \dfrac{\Pi_L + \Pi_R}{2} + a\dfrac{u_L - u_R}{2} \\ \rho_R^* &= \left(\dfrac{1}{\rho_R} + \dfrac{u_R - u^*}{a}\right)^{-1} \\ \mathbf{V}_R^* &= \mathbf{V}_R \end{cases} \tag{4.21}$$

and the Riemann solution is given by:

$$\mathcal{U}^R\left(\mathbf{U}_i, \mathbf{U}_{i+1}, x/t\right) = \begin{cases} \mathbf{U}_L = (u, \Pi, \rho, \mathbf{V}^T)_L^T, & \text{if} \quad x/t \leqslant u_L - \dfrac{a}{\rho_L}, \\ \mathbf{U}_L^* = (u, \Pi, \rho, \mathbf{V}^T)_L^{T,*}, & \text{else if} \quad x/t \leqslant u^*, \\ \mathbf{U}_R^* = (u, \Pi, \rho, \mathbf{V}^T)_R^{T,*}, & \text{else if} \quad x/t \leqslant u_R + \dfrac{a}{\rho_R}, \\ \mathbf{U}_R = (u, \Pi, \rho, \mathbf{V}^T)_R^T, & \text{else.} \end{cases}$$

Flux formula:

Now that the intermediate states of the Riemann problem are known, let us go back to the initial set of variables: $\mathbf{W} = \left(u, \rho, \mathbf{V}^T\right)^T$. To each of the intermediate states $\mathbf{U}_k$ of the Riemann solution for the relaxation scheme, we associate an intermediate state $\mathbf{W}_k = f(\mathbf{U}_k)$. f is the projection of $\mathbf{U}$ over $\mathbf{W}$ such that the values of ρ, Y, $\mathbf{u}$ and $\mathbf{V}$ are kept unchanged. Two flux formulae can now be considered:



(a) Riemann problem for the initial system.

(b) Riemann problem for the relaxation system.

Figure 4.2: *Sketch of the wave structure of the two Riemann problems that correspond to the initial non-linear system (on the left), with the left and right genuinely nonlinear waves, and the relaxation system (on the right), where the left and right waves are now contact discontinuities.*

- the first one uses the Riemann solution:

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}\left(f(\mathcal{U}^R(\mathbf{W}_i, \mathbf{W}_{i+1}, 0))\right), \tag{4.22}$$

- the other possibility consists in integrating the Riemann problem between t and $t+\Delta t$, and in space once from $-L$ to 0, and once from 0 to L, where $L > \max\left(u_L - \frac{a}{\rho_L}, (u_R + \frac{a}{\rho_R}\right)\Delta t$. This gives the formula:

$$\mathbf{F}_{i+\frac{1}{2}} = \frac{\mathbf{F}(\mathbf{W}_L) + \mathbf{F}(\mathbf{W}_R)}{2} - \frac{1}{2}\sum_k |j_k|(\mathbf{W}_{k+1} - \mathbf{W}_k) \tag{4.23}$$

  where the j_k are the wave velocities ($j_k \in \{u_L - a/\rho_L, u^*, u_R + a/\rho_R\}$) and $\mathbf{W}_k$ the successive intermediate states.

These formulae are equivalent in the case of Suliciu's method.

Scheme properties:

Let us check here whether the relaxation scheme verifies the three properties presented in 4.1.2.

Consistency: if $\mathbf{W}_L = \mathbf{W}_R = \overline{\mathbf{W}}$, then we clearly have from formula (4.23) that $\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}(\overline{\mathbf{W}})$, which proves that the flux is consistent.

Preservation of invariant domains: let Ω be the convex invariant domain for variable $\mathbf{W}$: it is made of all positive values for ρ, all values of Y in $[0,1]$ and each component of $\mathbf{V}$ satisfying a maximum principle. Then, we can extend it to a space Ω^ϵ for the variable $\mathbf{U}$, and verify that

$$\mathbf{U} \in \Omega^\epsilon \quad \Rightarrow \quad \mathbf{W} = f(\mathbf{U}) \in \Omega.$$

Now, as the Riemann solver is exact for variable $\mathbf{U}$, the scheme can be seen since a Godunov scheme for the relaxation system, that preserves the invariant domain for $\mathbf{U}$ under the CFL condition:

$$\Delta t \leqslant \frac{1}{2}\min_i \frac{\Delta x_i}{|\sigma_{i,j}|}, \quad \text{where} \quad |\sigma_{i,j}| = \max_{k=i,j}\left(|u_k| + \frac{a_{i,j}}{\rho_k}\right), \tag{4.24}$$

and $a_{i,j}$ are given by formula (4.16). Consequently, under the same CFL constraint, the invariant domain is also preserved for $\mathbf{W}$.

Entropy: as for invariant domains, the exact Riemann solver for $\mathbf{U}$ is entropic under the CFL constraint (4.24), because it is an exact Riemann solver for the relaxation system. According to arguments from Bouchut (2004), one can show that there also exists a discrete entropy inequality for the entropy η of the initial system, provided the CFL constraint and the subcharacteristic conditions are satisfied.

Vacuum and two-phase problems:

The relaxation scheme of Suliciu is robust, but in some configurations the subcharacteristic condition leads to a large overestimation of the acoustic wave velocities, that is used in the CFL constraint. This overestimation is due to strong variations of the density and of the sound velocity. Let us illustrate this by a numerical example:

Variable	Left state	Right state
density	$\rho_L = 1.0$	$\rho_R = 1.0 \cdot 10^3$
sound velocity	$c_L = 3.4 \cdot 10^2$	$c_R = 1.5 \cdot 10^3$
matter velocity	$u_L = 0.0$	$u_R = 0.0$

Table 4.1: *Two-phase flow configuration, where an interface separates a left gas phase from a right liquid phase.*

Then one has: $a_L = 3.4 \cdot 10^2$ and $a_R = 1.5 \cdot 10^6$. The time step is computed using the CFL constraint (4.24), which, in the case of Table 4.1, comes down to:

$$\Delta t \leqslant \frac{0.5\,\Delta x}{1.5 \cdot 10^6}.$$

This seems to indicate that the waves may propagate up to a velocity of $1.5 \cdot 10^6$ m/s. However, the maximum physical value of the acoustic waves is $c_R = 1.5 \cdot 10^3$. Consequently, the time step Δt could in reality be multiplied by 1000. While the HLLC scheme (Toro (2009)) accounts for this correct estimation, as we will see later, Suliciu's scheme does not permit it in this configuration.

In practice, this overestimated CFL constraint has two negative effects:

- the cost of the computation is multiplied by 1000 because the times steps are 1000 times too small,
- in certain flow regimes, as in low-Mach number flows, the value of the time step has an influence on the numerical diffusion: numerical experiments have shown that Suliciu's scheme gives very bad results comparing with the HLLC scheme.

The example presented above may be encountered in two flow configurations:

- in compressible single-phase flows (Euler equations), near a strong depression, where a state close to vacuum is generated. In this case, both density and acoustic velocity decrease locally,
- in two-phase flows at the interface between the two phases. The values that were set in the table correspond indeed to an air/water interface.

In the next section, we introduce two possible cures to this difficulty. The first one is a modification of Suliciu's scheme, while the second one is the HLLC scheme (Toro (2009)).

4.2.2 Bouchut's correction to vacuum

In Bouchut (2004), a correction to Suliciu's scheme is proposed in order to better handle vacuum situations in flows described by the Euler equations. Considering the relaxation system (4.12), the idea is to add a new equation: this time, this new equation is related to the variable a. This equation is a transport law that reads:

$$\partial_t a + u\partial_x a = 0.$$

The new system to be solved then writes:

$$\left\{\begin{array}{ccccc} \partial_t(\rho u) & + & \partial_x\left(\rho u^2 + \Pi\right) & = & 0 \\ \partial_t\left(\dfrac{\rho\Pi}{a^2}\right) & + & \partial_x\left(\dfrac{\rho\Pi}{a^2}u + u\right) & = & \rho\dfrac{P-\Pi}{a^2\epsilon} \\ \partial_t\rho & + & \partial_x(\rho u) & = & 0 \\ \partial_t(\rho a) & + & \partial_x(\rho a u) & = & 0 \\ \partial_t(\rho \mathbf{V}) & + & \partial_x(\rho \mathbf{V} u) & = & 0 \end{array}\right. . \tag{4.25}$$

This system is still hyperbolic with three eigenvalues that are:

$$\lambda_1 = u - a/\rho\,, \quad \lambda_2 = u + a/\rho \quad \text{and} \quad \lambda_{2+i} = u\,, \quad i = 1, ..., N-1$$

and, for the new vector of variables $\mathbf{Q} = \left(u, \Pi, \rho, a, \mathbf{V}^T\right)^T$, the eigenvectors associated with the eigenvalues are:

$$\mathbf{P}_1 = \left(1, -a, -a^2/\rho, \mathbf{0}\right)^T\,, \quad \mathbf{P}_2 = \left(1, a, a^2/\rho, \mathbf{0}\right)^T\,, \quad \mathbf{P}_{2+i} = \left(0, 0, \left(\frac{\delta_{i=j}}{\gamma_j}\right)^T_{j\leqslant N+2}\right)^T.$$

It is easy to check that the fields associated with the eigenvalues are all linearly degenerate.

The difference now is that a is given by the left and right states and is no longer the maximum value of these two states. The choice for evaluating a is discussed below.

Using the invariance of u and Π across the wave associated with λ_3, the invariance of a, $\mathbf{V}$ and $u \pm a/\rho$ across the waves associated with $\lambda_{1,2}$ and using the jump relations across these waves, one can find that the intermediate states are defined by:

$$\begin{cases} u^* &= \dfrac{a_L u_L + a_R u_R + \Pi_L - \Pi_R}{a_L + a_R} \\ \Pi^* &= \dfrac{a_R \Pi_L + a_L \Pi_R - a_L a_R (u_R - u_L)}{a_L + a_R} \\ \rho_L^* &= \left(\dfrac{1}{\rho_L} + \dfrac{a_R(u_R - u_L) + \Pi_L - \Pi_R}{a_L(a_L + a_R)}\right)^{-1} \\ a_L^* &= a_L \\ \mathbf{V}_L^* &= \mathbf{V}_L \end{cases} \quad \text{and} \quad \begin{cases} u^* &= \dfrac{a_L u_L + a_R u_R + \Pi_L - \Pi_R}{a_L + a_R} \\ \Pi^* &= \dfrac{a_R \Pi_L + a_L \Pi_R - a_L a_R (u_R - u_L)}{a_L + a_R} \\ \rho_R^* &= \left(\dfrac{1}{\rho_R} + \dfrac{a_L(u_R - u_L) + \Pi_R - \Pi_L}{a_R(a_L + a_R)}\right)^{-1} \\ a_R^* &= a_R \\ \mathbf{V}_R^* &= \mathbf{V}_R \end{cases} \tag{4.26}$$

The last point is to have a good estimation of a_L and a_R in the Riemann problem, to ensure that:

- the densities remain positive: $\rho_L^* > 0$ and $\rho_R^* > 0$,
- the scheme satisfies a discrete entropy inequality.

It can be done by looking for the lowest possible values for a_L and a_R in each Riemann problem, so that the two previous properties are satisfied. An explicit formula is given in Bouchut (2004), provided that the following assumptions are satisfied:

1. convexity of the pressure law according to $1/\rho$:

$$\forall \rho > 0, \quad \frac{d}{d\rho}\left(\rho\sqrt{p'(\rho)}\right) > 0,$$

2. $\rho\sqrt{p'(\rho)} \to \infty \quad \text{when} \quad \rho \to \infty$,
3. $\exists \beta \geqslant 1 \quad \text{s. t.} \quad \frac{d}{d\rho}\left(\rho\sqrt{p'(\rho)}\right) \leqslant \beta\sqrt{p'(\rho)}$.

Note that $\sqrt{p'(\rho)} = c(\rho, \mathbf{V})$. Under these assumptions, a_L and a_R can be defined by:

$$\begin{aligned} &\text{if} \quad \Pi_R - \Pi_L \geqslant 0, \quad \text{then} \begin{cases} a_L &= \rho_L c(\rho_L, \mathbf{V}_L) &+ \beta\rho_L \left(\frac{\Pi_R - \Pi_L}{\rho_R c(\rho_R, \mathbf{V}_R)} + u_L - u_R\right)_+ \\ a_R &= \rho_R c(\rho_R, \mathbf{V}_R) &+ \beta\rho_R \left(\frac{\Pi_L - \Pi_R}{a_L} + u_L - u_R\right)_+ \end{cases} \\ &\text{if} \quad \Pi_R - \Pi_L \leqslant 0, \quad \text{then} \begin{cases} a_L &= \rho_L c(\rho_L, \mathbf{V}_L) &+ \beta\rho_L \left(\frac{\Pi_R - \Pi_L}{a_R} + u_L - u_R\right)_+ \\ a_R &= \rho_R c(\rho_R, \mathbf{V}_R) &+ \beta\rho_R \left(\frac{\Pi_L - \Pi_R}{\rho_L c(\rho_L, \mathbf{V}_L)} + u_L - u_R\right)_+ \end{cases} \end{aligned} \tag{4.27}$$

with $(\cdot)_+$ being the positive part operator. With these formulae, positiveness of the intermediate values for density ρ_L^* and ρ_R^* and a discrete entropy inequality are ensured. They are well-suited for barotropic Euler equations, with a gamma pressure law.

With Bouchut's scheme, the CFL restriction becomes:

$$\frac{v_{\max}\Delta t}{\Delta x} \leqslant 1, \quad v_{\max} = \max_j \left(\left| u_j - \frac{a_j}{\rho_j} \right|, \left| u_j + \frac{a_j}{\rho_j} \right| \right). \tag{4.28}$$

We can note from formulae (4.27) that in the situation when the assumptions 1., 2. and 3. presented above are verified, the value of a is not far from the value of ρc. Thus, in configurations with an interface between two phases, one would not have the overestimation of the sound velocity anymore, which allows the time step to be substantially larger. In practice, it was observed that the time step is almost 1000 times larger with Bouchut's correction than with Suliciu's relaxation scheme, in the presence of a two-phase interface (defined in Table 4.1). The conditions 1., 2. and 3. proposed in Bouchut (2004) are applicable to barotropic Euler equations: pressure p depends on volume fraction ρ only. In the case of our two-phase systems, the state laws are different, since the mixture pressure P depends also on some variables from $\mathbf{V}$, such as Y, α and w. We will see, in the next section, that the same conditions for the two-phase systems are not always satisfied. The adaptation of Bouchut's correction to these systems necessitates further investigation, and the conditions 1., 2. and 3. could probably be adapted. This has not been done yet, and in the following we will adopt a different strategy. In particular, when the computation of the intermediate states leads to values out of the set of admissible values $\mathbf{\Omega}$, then we can consider an HLLC approximate solver as can be found in Toro (2009). There exists different forms of HLLC solvers: they differ by the values which are chosen to compute the wave velocities.

Remark 17. *In Bouchut (2004), it is noticed that the corrected relaxation scheme can be identified to an HLLC solver with a specific choice for the wave velocities, that are a_L, u^* and a_R.*

4.2.3 HLLC flux and final strategy

HLLC is a simple Riemann solver, made of four states separated by three discontinuities that are: S_L, S^* and S_R. Let us introduce $j_L = \rho_L (u_L - S_L)$ and $j_R = \rho_R (u_R - S_R)$. The intermediate states are given by:

$$\begin{cases} u_L^* = S^* &= \dfrac{u_R j_R - u_L j_L}{j_R - j_L} + \dfrac{p_R - p_L}{j_R - j_L} \\ p_L^* = p^* &= \dfrac{p_L j_R - p_R j_L}{j_R - j_L} + j_L j_R \dfrac{u_L - u_R}{j_R - j_L} \\ \rho_L^* &= \rho_L \dfrac{u_L - S_L}{S^* - S_L} \\ \mathbf{V}_L^* &= \mathbf{V}_L \end{cases} \quad \text{and} \quad \begin{cases} u_R^* = S^* &= \dfrac{u_R j_R - u_L j_L}{j_R - j_L} + \dfrac{p_R - p_L}{j_R - j_L} \\ p_R^* = p^* &= \dfrac{p_L j_R - p_R j_L}{j_R - j_L} + j_L j_R \dfrac{u_L - u_R}{j_R - j_L} \\ \rho_R^* &= \rho_R \dfrac{u_R - S_R}{S^* - S_R} \\ \mathbf{V}_R^* &= \mathbf{V}_R \end{cases} \tag{4.29}$$

In the literature, one can find multiple estimates for S_L and S_R. We choose here to use an estimation proposed in Davis (1988) and slightly modified in Toro (2009), which writes:

$$\begin{aligned} S_L &= -\max\left(|u_L| + c_L, |u_R| + c_R\right), \\ S_R &= \max\left(|u_L| + c_L, |u_R| + c_R\right). \end{aligned}$$

As for Bouchut's corrected scheme, the characteristic velocities for the HLLC scheme are not as restrictive as those of Suliciu's relaxation, since they are closer to the exact value of ρc.

Finally, we choose to apply the strategy of Figure 4.3 to compute the numerical fluxes of system (4.9). Let us name Relax-S Suliciu's relaxation scheme adapted to the two-phase systems of equations, Relax-B the corrected scheme proposed in Bouchut (2004) and adapted to our systems.

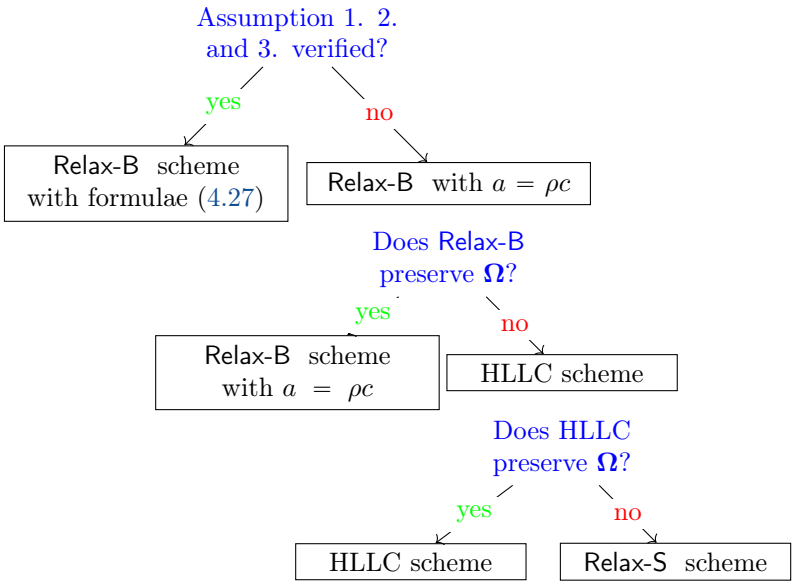


Figure 4.3: *Sketch of the resolution strategy for the numerical flux computation. Ω is the invariant space.*

4.2.4 Resolution of the transport equations and preservation of uniform velocity and pressure profiles

The last point we have to study in this section, concerns the update formulae of some of the variables of $\mathbf{V}$. Indeed, these variables are transported and not conserved. Until now, we have written the transport equations in their conservative form and ensured that the set of admissible values $\mathbf{\Omega}$ is preserved. Now, some other properties should be verified for the variables $\mathbf{V}$.

First, we want these variables to satisfy a **maximum principle**, meaning that they are bounded throughout the simulation. Let v be one of these variables. From time t^n to time t^{n+1}, one is looking for the following property:

$$\forall i, \min_j(v(t^n, x_j)) \leqslant v(t^{n+1}, x_i) \leqslant \max_j(v(t^n, x_j)).$$

We may even require a local maximum principle. In 1D this reads:

$$\forall i\,, \quad v(t^{n+1}, x_i) \in [\min_{k\in\{i-1,i,i+1\}} v(t^n, x_k), \max_{k\in\{i-1,i,i+1\}} v(t^n, x_k)]. \tag{4.30}$$

Another aspect, which is related to the specific two-phase systems is that initially uniform velocity and pressure fields must remain uniform throughout the computation. It is known that a bad discretization of the volume fraction equation may raise difficulties to keep these uniform profiles, and in general cases, may generate instabilities (see an overview and a solution to this problem in Abgrall (1996)). In the following, we will then verify this property for the conservative discretization, as formulated in the previous section, and for an upwind discretization. We will moreover impose that the phasic pressures p_k, if they exist, are also uniform and equal: $p_k = \overline{p}$. This means that the scheme must be able to preserve a pressure equilibrium. Let assume that for all the systems under study: *there exists a conservative variable m, a non-conservative one α belonging to the vector $\mathbf{V}$, and a bijective function of the pressure $q : p \mapsto q(p)$ such that m writes:*

$$m = \alpha q(p) \tag{4.31}$$

Remark 18. *Relation (4.31) looks like a pressure law. This form of equation has been chosen since it corresponds to the form of the phasic pressure laws for the two-fluid systems of equations. Here, it will be useful to perform generic analyses.*

Update using a conservative scheme

Let us first see if the maximum principle is satisfied when the conservative variable ρv is updated using the approximate Riemann solvers as presented previously. From time t^n to time $t^{n+1} = t^n + \Delta t$, the update formula writes:

$$\begin{aligned}(\rho v)_i^{n+1} &= (\rho v)_i^n + \frac{\Delta t}{\Delta x}\left[\mathbf{F}\left((\rho v)^{\mathcal{R}}_{i-\frac{1}{2}}\right) - \mathbf{F}\left((\rho v)^{\mathcal{R}}_{i+\frac{1}{2}}\right)\right] \\ &= \rho_i^n v_i^n + \rho^*_{i-\frac{1}{2}} u^*_{i-\frac{1}{2}} \frac{\Delta t}{\Delta x} v^*_{i-\frac{1}{2}} - \rho^*_{i+\frac{1}{2}} u^*_{i+\frac{1}{2}} \frac{\Delta t}{\Delta x} v^*_{i+\frac{1}{2}}\end{aligned} \tag{4.32}$$

and knowing that the formula for updating ρ writes:

$$\rho_i^{n+1} = \rho_i^n + \rho^*_{i-\frac{1}{2}} u^*_{i-\frac{1}{2}} \frac{\Delta t}{\Delta x} - \rho^*_{i+\frac{1}{2}} u^*_{i+\frac{1}{2}} \frac{\Delta t}{\Delta x}, \tag{4.33}$$

one can see that when dividing (4.32) by (4.33), v_i^{n+1} writes as a convex combination of v_i^n, $v^*_{i-\frac{1}{2}}$ and $v^*_{i+\frac{1}{2}}$. Then, the maximum principle is satisfied.

Now, let us assume that at time t^n, there is a uniform pressure field $\overline{p}$ (with $p_1 = p_2 = \overline{p}$) and a uniform velocity field $\overline{u} > 0$. Then, one can show that the update relation for a conservative variable m, such as the equation (4.33), becomes:

$$m_i^{n+1} = m_i^n + m_{i-1}^n \overline{u} \frac{\Delta t}{\Delta x} - m_i^n \overline{u} \frac{\Delta t}{\Delta x}. \tag{4.34}$$

On the other hand, the equation for α is:

$$(\rho\alpha)_i^{n+1} = (\rho\alpha)_i^n + \overline{u} \frac{\Delta t}{\Delta x} (\rho\alpha)_{i-1}^n - \overline{u} \frac{\Delta t}{\Delta x} (\rho\alpha)_i^n . \tag{4.35}$$

Combining these two formulae, the update formula of $q(p)$ reads (we note $q_i^n = q(p_i^n)$):

$$\begin{aligned} q_i^{n+1} &= \frac{m_i^{n+1}}{\alpha_i^{n+1}} \\ &= \frac{m_i^{n+1} \rho_i^{n+1}}{(\rho\alpha)_i^{n+1}} \\ &= \frac{q_i^n \left(\rho_i^n + \overline{u}\frac{\Delta t}{\Delta x}\left(\rho_{i-1}^n - \rho_i^n\right)\right)\left(\alpha_i^n + \overline{u}\frac{\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right)\right)}{(\rho\alpha)_i^n + \overline{u}\frac{\Delta t}{\Delta x}\left((\rho\alpha)_{i-1}^n - (\rho\alpha)_i^n\right)} \end{aligned}$$

and we can see from the last line, that in most cases, we will not have equality between q_i^{n+1} and q_i^n. Consequently, the pressure value is not kept constant, and according to Abgrall (1996), instabilities may appear in the numerical solution.

Update using an upwind discretization

Discretizations for the non-conservative variables are proposed in many numerical schemes. We consider here an upwind scheme inspired from Godlewski and Raviart (1996); Kokh and Lagoutière (2010), which reads:

$$v_i^{n+1} = v_i^n + \frac{\Delta t}{\Delta x}\left((vu)_{i-\frac{1}{2}} - (vu)_{i+\frac{1}{2}}\right) - v_i^n \frac{\Delta t}{\Delta x}\left(u_{i-\frac{1}{2}} - u_{i+\frac{1}{2}}\right). \tag{4.36}$$

Equation (4.36) can also be written as:

$$\begin{aligned} v_i^{n+1} &= v_i^n \left(1 - u_{i-\frac{1}{2}} \frac{\Delta t}{\Delta x}\right) + u_{i-\frac{1}{2}} \frac{\Delta t}{\Delta x} v_{i-\frac{1}{2}} \\ &\quad - v_i^n \left(1 - u_{i+\frac{1}{2}} \frac{\Delta t}{\Delta x}\right) - u_{i+\frac{1}{2}} \frac{\Delta t}{\Delta x} v_{i+\frac{1}{2}} \\ &\quad + v_i^n \end{aligned}$$

and one can show that, whatever the values of $u_{i-\frac{1}{2}}$ and $u_{i+\frac{1}{2}}$, this last formula is a convex combination of v_i^n, v_{i-1}^n and v_{i+1}^n. Hence, formula (4.36) satisfies a local maximum principle.

Let us assume now that we have (4.31) and let us look at the updated value of q in the case of uniform velocity and pressure fields ($\overline{u} > 0$):

$$\begin{aligned} q_i^{n+1} &= \frac{m_i^{n+1}}{\alpha_i^{n+1}} \\ &= \frac{q_i^n \left(\alpha_i^n + \overline{u}\frac{\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right)\right)}{\alpha_i^n + \overline{u}\frac{\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right)} \\ &= q_i^n \end{aligned}$$

There, the pressure profile is kept constant.

Remark 19. *Formula (4.36) has been modified in Kokh and Lagoutière (2010) to get a numerical scheme with very little diffusion on the transported variables. This is performed by noting that formula (4.36) is in fact an upwind numerical scheme, which is stable under a CFL condition. Fully downwind schemes are not diffusive, but not stable either. For each system of equations, it is then possible to get a balance between upwind and downwind fluxes, that maintains the stability of the solution while reducing the numerical diffusion.*

4.2.5 Summary

To conclude this part, let us give the main features of the whole numerical scheme we will use in the following chapters.

1. spatial and temporal discretization are of first order. The finite volume formulation for the conservative variables is given by formula (4.4), which is a standard finite volume formulation (see for instance in Leveque (2004)),
2. the numerical flux is obtained by a relaxation method, similar to the Suliciu's relaxation scheme (Suliciu (1990)), named Relax-S, with a correction inspired by Bouchut (2004) and applied to the two-phase system (4.9): Relax-B scheme. Since all stability properties have not been proved yet, the application of the scheme proposed in Bouchut (2004) to our systems of equations could raise some difficulties. In those cases, we choose either the HLLC numerical scheme, which is close to Relax-B, or Relax-S scheme, according to the procedure of Figure 4.3,
3. the discretization of some of the transported variables (α and w for the two-phase equations) is given by the upwind formula (4.36) as it is done in Godlewski and Raviart (1996); Kokh and Lagoutière (2010),
4. the time step Δt is computed so as to satisfy the appropriate CFL condition.

The use of Relax-B is an original contribution in the domain of two-phase flows, where Suliciu's relaxation or other relaxation schemes are usually used.

4.3 Application of the strategy to the two-phase systems of equations

Now that the general form of the numerical scheme has been presented, let us see how it can be applied to each of the three systems of equations (4.37), (4.40) and (4.41).

4.3.1 Discretization of the 3-equation system

In this part, we consider the 3-equation system (4.37). In 1D, it writes:

$$\left\{\begin{array}{rcccc} \partial_t \rho & + & \partial_x(\rho u) & = & 0 \\ \partial_t(\rho Y) & + & \partial_x(\rho Y u) & = & 0 \\ \partial_t(\rho u) & + & \partial_x\left(\rho u^2 + p\right) & = & 0 \\ p = p_1(\rho, Y) = p_2(\rho, Y) & & & & \end{array}\right. \tag{4.37}$$

Let us also recall some properties of this system:

- **Pressure equality:** in system (4.37), one assumes a mechanical equilibrium, meaning that $p_1 = p_2$. Consequently, in the momentum equation, the pressure value is also given by this equality: $p = p_1 = p_2$, and one can compute the volume fractions of the phases using the pressure laws and:

$$p_1\left(\frac{\rho Y}{\alpha_1}\right) = p_2\left(\frac{\rho(1-Y)}{\alpha_2}\right),$$

  with $\alpha = \alpha_1 = 1 - \alpha_2$. Details are given in paragraph 4.3.1.2.

- **Transport equations:** from the generic formulation (4.9), the vector $\mathbf{V}$ is composed of Y, u_y and u_z. Moreover, the mass fraction Y multiplied by the total density ρ gives the partial mass of the first fluid m_1, which is a conserved variable.

- **Wood's sound velocity:** the characteristic sound velocity of system (4.37) is Wood's sound velocity, which writes using the barotropic pressure laws:

$$\mathrm{c}_{\mathrm{Wood}}(\rho, Y) = \left(\frac{\alpha(\rho,Y)^2}{Y c_1^2} + \frac{(1-\alpha(\rho,Y))^2}{(1-Y)c_2^2}\right)^{-1}.$$

4.3.1.1 An adapted numerical scheme

Now, we would like to know whether Suliciu's relaxation scheme with Bouchut's correction, *i.e.* Relax-B scheme, can be applied directly, meaning that the assumptions 1. 2. and 3. from paragraph 4.2.2 are satisfied.

Let us introduce an auxiliary function $X(\rho)$ such that:

$$\begin{aligned} \sqrt{p'(\rho)} &= \mathrm{c}_{\mathrm{Wood}}(\rho, Y) \\ &= (X(\rho))^{-1/2}. \end{aligned}$$

Then, one can compute:

$$\begin{aligned} \frac{\partial}{\partial\rho}\left(\rho\sqrt{p'(\rho)}\right) &= \mathrm{c}_{\mathrm{Wood}}\left(1 + \frac{\rho}{\mathrm{c}_{\mathrm{Wood}}}\frac{\partial \mathrm{c}_{\mathrm{Wood}}}{\partial\rho}\right) \\ &= \mathrm{c}_{\mathrm{Wood}}\left(1 - \frac{\rho}{2X}\frac{\partial X}{\partial\rho}\right) \\ &= \mathrm{c}_{\mathrm{Wood}}\left(1 - \frac{\rho}{2X}\left[\frac{2\alpha}{Y c_1^2} - \frac{2(1-\alpha)}{(1-Y)c_2^2}\right]\frac{\partial\alpha}{\partial\rho}\right). \end{aligned}$$

Using pressure equality and the barotropic pressure laws, one can show that:

$$\frac{\partial\alpha}{\partial\rho} = \frac{\alpha(1-\alpha)}{\rho}\frac{(1-\alpha)Y c_1^2 - \alpha(1-Y)c_2^2}{(1-\alpha)^2 Y c_1^2 + \alpha^2(1-Y)c_2^2}.$$

Replacing it in the above relation gives:

$$\frac{\partial}{\partial\rho}\left(\rho\sqrt{p'(\rho)}\right) = \mathrm{c}_{\mathrm{Wood}}\left(1+\frac{\alpha(1-\alpha)}{X}\frac{\left((1-\alpha)Yc_1^2-\alpha(1-Y)c_2^2\right)^2}{((1-\alpha)^2Yc_1^2+\alpha^2(1-Y)c_2^2)\,Y(1-Y)c_1^2c_2^2}\right)$$
$$= \mathrm{c}_{\mathrm{Wood}}\left(1+\alpha(1-\alpha)\left(\frac{(1-\alpha)Yc_1^2-\alpha(1-Y)c_2^2}{(1-\alpha)^2Yc_1^2+\alpha^2(1-Y)c_2^2}\right)^2\right).$$

The first assumption, $\frac{\partial}{\partial\rho}\left(\rho\sqrt{p'(\rho)}\right) > 0$, is clearly verified.

One can also easily show that the second assumption is satisfied, by noting that $\mathrm{c}_{\mathrm{Wood}}$ is bounded for $\alpha \in [0,1]$ (for a fixed value of mass fraction Y), and $\mathrm{c}_{\mathrm{Wood}} > m$, with $m > 0$.

For the third assumption, let us define the new function $f : \alpha \mapsto f(\alpha)$ such that

$$f(\alpha) = \alpha(1-\alpha)\left(\frac{(1-\alpha)B_1-\alpha B_2}{(1-\alpha)^2B_1+\alpha^2B_2}\right)^2,$$

where constants B_1 and B_2 are positive (and correspond respectively to Yc_1^2 and $(1-Y)c_2^2$). One can show that f is defined, continuous and positive over $[0,1]$, and $f(0) = 0$, $f(1) = 1$. Thus, there exists a maximum value of f within this interval, that depends on the value of the parameters B_1 and B_2, and the third assumption is also satisfied.

Studying the variations of f and finding its maximum value in $[0,1]$ is quite complex. Let us just find an upper bound. Let us consider the function $g : \alpha \mapsto g(\alpha)$ defined by:

$$g(\alpha) = \frac{(1-\alpha)B_1-\alpha B_2}{(1-\alpha)^2B_1+\alpha^2B_2} = \frac{a(\alpha)}{b(\alpha)}.$$

One has:

$$|a(\alpha)| \leqslant \max(B_1,B_2) \leqslant B_1+B_2,$$

and one can show that:

$$b(\alpha) \geqslant \frac{B_1B_2}{B_1+B_2},$$

which leads to:

$$|g(\alpha)| \leqslant \frac{(B_1+B_2)^2}{B_1B_2},$$

and finally:

$$f(\alpha) \leqslant \frac{(B_1+B_2)^4}{(B_1B_2)^2}.$$

For the 3-equation system, one choice for β is then:

$$\beta_{3\mathrm{eq}} = 1+\frac{(Yc_1^2+(1-Y)c_2^2)^4}{(Y(1-Y)c_1^2c_2^2)^2}. \tag{4.38}$$

In practice, the estimation (4.38) is very coarse, and largely overestimates the real coefficient β, especially in cases where $Y(1-Y) \ll 1$: $\beta_{3\mathrm{eq}}$ may become arbitrarily large, in particular near zones of pure phases. We can thus wonder whether the three conditions 1., 2. and 3. as proposed in Bouchut (2004) should not be adapted to the pressure laws we are using here. We propose however to apply Relax-B and change to HLLC or Relax-S when the computed values of ρ and ρY are out of $\mathbf{\Omega}$.

4.3.1.2 Closure relations and equation on α and p

Once ρ, ρY and ρu are updated using the finite volume scheme, the second step for solving system (4.37) is to update the pressure variable p.

Following the lines of Chanteperdrix, Villedieu, and Vila (2002), a simple way consists in first computing α from the equality:

$$p_1\left(\frac{\rho Y}{\alpha}\right) = p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right),$$

which, when using linearized pressure laws in the form:

$$p_k\left(\rho_k\right) = p_k^o + c_k^2(\rho_k - \rho_k^o),$$

comes down to solve a second-order polynomial equation of the form:

$$\alpha^2 q^o + (q - q^o)\alpha - \rho Y c_1^2 = 0\,, \quad \text{with} \quad \begin{cases} q^o &= \rho_2^o c_2^2 &- &\rho_1^o c_1^2\,, \\ q &= \rho Y c_1^2 &+ &\rho(1-Y)c_2^2\,. \end{cases} \tag{4.39}$$

There exists a solution of (4.39) in $[0,1]$ and this solution is unique (see Chanteperdrix, Villedieu, and Vila (2002)). Pressure is finally computed through p_1 or p_2 pressure laws:

$$p = p_1\left(\frac{\rho Y}{\alpha}\right) = p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right).$$

4.3.1.3 Uniform pressure and velocity profiles

In the 3-equation system, no equation is solved by using the advection discretization. Let us see whether Relax-B preserves uniform velocity and pressure profiles.

Let us assume that at time t^n, one has $u(t^n, x_i) = \overline{u} > 0$ and $p(t^n, x_i) = \overline{p}$. Then at time $t^{n+1} = t^n + \Delta t$, we will have:

$$\begin{aligned}
(\rho u)_i^{n+1} &= (\rho u)_i^n + \frac{\Delta t}{\Delta x}\left(\rho_{i-1}^n \overline{u}^2 + \overline{p} - \rho_i^n \overline{u}^2 - \overline{p}\right) \\
&= \overline{u}\left[\rho_i^n + \frac{\overline{u}\Delta t}{\Delta x}\left(\rho_{i-1}^n - \rho_i^n\right)\right], \\
\rho_i^{n+1} &= \rho_i^n + \frac{\Delta t}{\Delta x}\left(\rho_{i-1}^n \overline{u} - \rho_i^n \overline{u}\right), \\
(\rho Y)_i^{n+1} &= (\rho Y)_i^n + \frac{\Delta t}{\Delta x}\left(\rho_{i-1}^n Y_{i-1}^n \overline{u} - \rho_i^n Y_i^n \overline{u}\right),
\end{aligned}$$

and by dividing the first relation by the second one, one recovers $u_i^{n+1} = \overline{u}$.

Now, let us show that the uniform pressure profile is also preserved. At time t^n, we have the relation: $\rho_i^n Y_i^n = \alpha_i^n \overline{\rho_1}$ with $\overline{\rho_1}$ such that $p_1\left(\overline{\rho_1}\right) = \overline{p}$. Then the third update formula reads:

$$(\alpha\rho_1)_i^{n+1} = \overline{\rho_1}\left[\alpha_i^n + \frac{\overline{u}\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right)\right].$$

From $\rho_i^{n+1} - (\rho Y)_i^{n+1}$, we can have a similar relation for ρ_2, with $p_2\left(\overline{\rho_2}\right) = \overline{p}$, that writes:

$$((1-\alpha)\rho_2)_i^{n+1} = \overline{\rho_2}\left[1 - \alpha_i^n - \frac{\overline{u}\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right)\right].$$

Let us introduce $\tilde{\alpha}$ such that $\tilde{\alpha} = \alpha_i^n + \frac{\overline{u}\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right)$, and compute $\tilde{\rho_1} = \frac{(\alpha\rho_1)_i^{n+1}}{\tilde{\alpha}}$, $\tilde{\rho_2} = \frac{((1-\alpha)\rho_2)_i^{n+1}}{1-\tilde{\alpha}}$. It is obvious that $\tilde{\rho_1} = \overline{\rho_1}$ and $\tilde{\rho_2} = \overline{\rho_2}$. Moreover, we also have $p_1(\tilde{\rho_1}) = p_2(\tilde{\rho_2}) = \overline{p}$. Consequently, we can say that $\tilde{\alpha}$ is an equilibrium value of α at time t^{n+1}, and it preserves the uniform pressure field. As the equilibrium value is unique, we have necessarily $\alpha_i^{n+1} = \tilde{\alpha}$ and the property is proved.

4.3.1.4 Summary

Although the explicit formulation for the wave velocities in Relax-B does not directly apply for the 3-equation system, in practice this scheme has been used for the numerical tests presented at the end of this chapter and it is stable. In the 3-equation system, there is no non-conservative terms. The system is closed by the pressure equality. An equilibrium volume fraction can thus be computed, and the whole numerical scheme preserves uniform velocity and pressure profiles. The unique errors which have been observed are due to the choice of the linearized pressure laws. Indeed, these laws write:

$$
\begin{aligned}
p_1 &= p_1^o + c_1^2 \frac{\rho Y}{\alpha}, \\
p_2 &= p_2^o + c_2^2 \frac{\rho(1-Y)}{1-\alpha},
\end{aligned}
$$

with $c_1^2 = O(10^5)$ and $c_2^2 = O(10^6)$. Then, errors on the computation of α may be amplified by several orders of magnitude. These errors are observed in the numerical tests, but do not amplify during the simulations.

4.3.2 Discretization of the 4-equation system

The 4-equation system writes:

$$
\left\{
\begin{array}{rcccc}
\partial_t \rho & + & \partial_x (\rho u) & = & 0 \\
\partial_t (\rho Y) & + & \partial_x (\rho Y u) & = & 0 \\
\partial_t (\rho u) & + & \partial_x \left(\rho u^2 + p\right) & = & 0 \\
\partial_t \alpha & + & u \partial_x \alpha & = & \dfrac{p_1 - p_2}{\epsilon}
\end{array}
\right.
\qquad
p = \alpha p_1(\rho, Y, \alpha) + (1-\alpha) p_2(\rho, Y, \alpha)
\tag{4.40}
$$

Let us also recall some properties of this system:

- **Transport equations:** in this system, the vector $\mathbf{V}$ is composed of Y, α, u_y and u_z. As for the 3-equation system, the mass fraction is related to the conservation of the mass of one of the fluids. It is thus better to update the conservative variable ρY rather than Y. On the contrary, α has to be updated using the advection discretization, to keep uniform pressure and velocity profiles, as we will see in section 4.3.2.2.
- **Frozen sound velocity:** the characteristic sound velocity for the 4-equation system is given by the Frozen sound velocity:

$$
\mathrm{c}_{\mathrm{Frozen}}(\rho, Y, \alpha) = \sqrt{Y c_1^2 + (1-Y) c_2^2}.
$$

- **Pressure:** contrary to the 3-equation system, here pressures p_1 and p_2 are no longer equal. In the momentum equation, the mixture pressure P is now defined by

$$
P = \alpha p_1 \left(\frac{\rho Y}{\alpha} \right) + (1-\alpha) p_2 \left(\frac{\rho(1-Y)}{1-\alpha} \right).
$$

4.3.2.1 Adapted numerical scheme

For the 4-equation system, one can use Relax-B scheme with the wave velocities computed with (4.27). Indeed, it is clear that:

$$
\frac{\partial \left(\rho \sqrt{p'(\rho)} \right)}{\partial \rho} = \mathrm{c}_{\mathrm{Frozen}} > 0,
$$

and

$$\rho c_{\text{Frozen}} \xrightarrow[\rho\to\infty]{} \infty.$$

Finally, the coefficient β_{4eq} of the third assumption is given by $\beta_{4eq} = 1$. The fluxes for ρ, ρY and ρu can thus be computed using this flux.

4.3.2.2 Non-conservative equations discretization

The equation on α is discretized using (4.36). To check that the uniform velocity and pressure profiles are preserved, let us note that from formula (4.31), we have $m = \rho Y = \alpha\rho_1(P)$ and the demonstrations of paragraph 4.2.4 are valid.

We can also refer to the proof for the 3-equation system, by noting that we have explicitly the update relation:

$$\alpha_i^{n+1} = \alpha_i^n + \frac{\overline{u}\Delta t}{\Delta x}\left(\alpha_{i-1}^n - \alpha_i^n\right).$$

4.3.2.3 Source term and its effects on the solution

In the 4-equation system, the relaxation source term brings the solution closer to a mechanical equilibrium. By discarding this source term, numerical solutions can completely differ from the real physics. In particular, we will see that the solutions to simple Riemann problems differ between the 3-equation system and the 4-equation one, and the relaxation to an equilibrium solution of the 4-equation solution is not as efficient as solving directly the 3-equation system. Moreover, as the characteristic velocities are given by different formulae, except when one deals with pure phases, the temporal discretization may also vary between the two models. It would be interesting to compare the numerical scheme proposed in Chanteperdrix, Villedieu, and Vila (2002), where the 4-equation system is used as a relaxation system in order to solve the 3-equation one, with Relax-B scheme applied directly on the 3-equation system.

4.3.3 Discretization of the 5-equation system

The 5-equation system with both micro-inertia and micro-viscosity writes:

$$\left\{\begin{array}{lcll} \partial_t\rho & + & \partial_x(\rho u) & = 0 \\ \partial_t(\rho Y) & + & \partial_x(\rho Y u) & = 0 \\ \partial_t(\rho u) & + & \partial_x\left(\rho u^2 + \left(p + \frac{1}{2}(\rho Y w)^2\right)\right) & = 0 \\ \partial_t\alpha & + & u\partial_x\alpha & = \dfrac{\rho Y w}{\sqrt{\nu}} \\ \partial_t w & + & u\partial_x w & = \dfrac{p_1 - p_2}{\rho Y\sqrt{\nu}} - \dfrac{\epsilon}{\nu}w \end{array}\right. \tag{4.41}$$
$$p = \alpha p_1(\rho, Y, \alpha) + (1-\alpha)p_2(\rho, Y, \alpha)$$

In application to the generic approach, its specific properties are:

- **Transport equations:** the vector $\mathbf{V}$ is composed of the variables Y, α, w, u_y and u_z. Once more, as Y is related to the conservation of a partial mass, we solve it by using the conservative variable ρY.
- **Pulsation sound velocity:** the characteristic sound velocity is given by the formula:

$$c_{\text{Puls}}^2(\rho, Y, \alpha, w) = c_{\text{Frozen}}^2 + \rho\left(Yw\right)^2.$$

- **Pressure:** the pressure P appearing in the momentum equation accounts for an additional term issued from the pulsation energy. It writes:

$$P = \alpha p_1\left(\frac{\rho Y}{\alpha}\right) + (1-\alpha)p_2\left(\frac{\rho(1-Y)}{1-\alpha}\right) + \frac{1}{2}\left(\rho Y w\right)^2.$$

4.3.3.1 Adapted numerical scheme

Bouchut's explicit formulae for the value of the waves cannot be used here. Indeed, although the first two conditions are satisfied, namely:

$$\frac{\partial\left(\rho\sqrt{p'(\rho)}\right)}{\partial\rho} = \mathrm{c_{Puls}}\left(1+\rho\frac{(Yw)^2}{2\mathrm{c_{Puls}}}\right) > 0$$

and

$$\rho\mathrm{c_{Puls}} \xrightarrow[\rho\to\infty]{} \infty,$$

there does not exist any upper bound to $\rho\frac{(Yw)^2}{2\mathrm{c_{Puls}}}$, as w may take arbitrary large values. Thus, one cannot find any $\beta_{5\mathrm{eq}}$ to satisfy the third condition. As for the 3-equation system, we can wonder whether an adaptation of the conditions of Bouchut (2004) to the case of state laws accounting for w would enable to find more appropriate formulae and stability criteria. For now, we use Relax-B scheme assuming that $\beta_{5\mathrm{eq}} = 1$. If it fails, we will use HLLC scheme.

4.3.3.2 Non-conservative equations discretization

Here, the non-conservative variables are α and w. We choose to discretize them by using (4.36). The case of initial uniform velocity and pressure profiles is studied by assuming that the initial value of w is zero everywhere. Indeed, if vibrations occurred at some places, the pressure P in the momentum equation would not have a uniform profile, unless ρ and Y profiles are also uniform. Therefore, for more simplicity, we also impose that the schemes preserves $w = 0$.

Let us assume that the velocity is $u = \overline{u} > 0$ and pressure $p = \overline{p}$, then, w is updated by the equation:

$$w_i^{n+1} = w_i^n + \frac{\overline{u}\Delta t}{\Delta x}\left(w_{i-1}^n - w_i^n\right),$$

and if $w_i^n = 0$, then $w_i^{n+1} = 0$. The pressure function is thus reduced to the thermodynamic pressure $P = \alpha p_1 + (1-\alpha)p_2$, and using the same arguments as for the 4-equation system, the resolution of α by using (4.36) preserves a constant pressure profile (and pressure equilibrium).

4.3.3.3 Source terms and their effects on the solution

The source terms in the 5-equation system stand for micro-vibrations effects when $w \neq 0$. These vibrations are generated when there is a pressure disequilibrium: $p_1 - p_2 \neq 0$. Without these source terms, the variable w is only transported and as for the 4-equation system, the resolution of the convective part only has no physical meaning.

4.4 Some one-dimensional verification tests:

To conclude this chapter, we present here some 1D simulations, which aim at testing the Relax-B numerical scheme (and compare it with Relax-S). Some of these tests are inspired from classic Riemann problems for compressible fluids that can be found in the literature (Toro (2009)). They have been applied to the 3-equation system in Chanteperdrix, Villedieu, and Vila (2002) and similar systems in Kokh and Lagoutière (2010). The exact solutions are computed in Appendix A. As the source terms are not integrated, these tests do not present any physical interest. We will however see that the Riemann solutions for one system of equations or an other, can greatly vary. For one of the tests, we perform a projection of the solution to the 4-equation system on the solution with pressure equilibrium (we let ϵ tend to 0), following the strategy of Chanteperdrix, Villedieu, and Vila (2002) but with a different Riemann solver for the 4-equation system. In our case, we show the difference with the solution given by the scheme directly adapted for the 3-equation system.

In the following we do not present numerical results for the 5-equation system. Indeed, we have found no Riemann problem in the literature for such a system with micro-inertia. Moreover, initially uniform profiles for w remain uniform, and the resolution of the 5-equation system comes down to solve the 4-equation one. However, it would be interesting to study the effects of non zero gradients of w and see whether the numerical scheme handles these new configurations.

4.4.1 1D advection test case

The two-phase advection test aims at testing the preservation of uniform pressure and velocity profiles. *Infinite domain* boundary conditions are used. The composition of the two-phase mixture is given by the profile of volume fraction α, which is initially at equilibrium (satisfying $p_1 = p_2$). Moreover, we impose that a sharp profile initially separates the two phases. This sharp limit illustrates the advantages of using Relax-B scheme instead of Relax-S. The precise initial conditions are given in the tables below:

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$p_0 = 1.0 \cdot 10^5$	$p_0 = 1.0 \cdot 10^5$
mass fraction	$Y_L = 1.0 - 1.0 \cdot 10^{-8}$	$Y_R = 1.0 \cdot 10^{-11}$
velocity	$u_0 = 1.0$	$u_0 = 1.0$
volume fraction	$\alpha_L = 1.0 - 1.0 \cdot 10^{-8}$	$\alpha_R = 1.0 \cdot 10^{-8}$
vibration	$w_0 = 0.0$	$w_0 = 0.0$

On Figure 4.4, we show comparisons between the exact solution and the numerical solutions of the 3-equation system obtained with Relax-S and Relax-B. Both numerical solutions provide the same level of accuracy: the initially sharp interface is spread over a few cells due to the numerical diffusion. Velocity and pressure profiles are kept approximately uniform, except from small amplitude errors that are illustrated in Figure 4.5. These instabilities are in fact round-off errors due to the choice of the pressure laws, as explained previously. However they stay bounded throughout the simulation and do not increase neither when the spatial discretization is finer, as the comparisons of Figure 4.5 show.

The values of the time steps computed for Relax-S and Relax-B are compared in Figure 4.6. They are given in function of the simulated physical time. One can see that in the initial time steps, the value of Δt for Relax-S is about 20 times smaller than Δt computed for Relax-B. This is due to the high ratio of densities and acoustic velocities on both sides of the interface. Indeed, we have used the following values, that are close to physical values for air and water:

$$\rho_1 = 1.0 \text{ kg/m}^3, \quad \rho_2 = 1.0 \cdot 10^3 \text{ kg/m}^3, \quad c_1 = 3.4 \cdot 10^2 \text{ m/s}, \quad c_2 = 1.5 \cdot 10^3 \text{ m/s}.$$

Then, due to the numerical diffusion and the creation of a mixture zone, this ratio decreases. We also present in Figure 4.6 the number of time iterations needed to perform the whole computation, by both Relax-S and Relax-B schemes. At the end of the simulation, we clearly see that using Relax-B scheme requires 3 times less time iterations than Relax-S. Moreover, Relax-B scheme is not more complex than Relax-S scheme and their numerical solutions are of the same order of accuracy (at least for the advection test). Relax-B seems then more advantageous than Relax-S.

The results for the 4-equation solution obtained with Relax-B scheme are presented in Figure 4.7. As for the 3-equation resolution, the computation of the pressure leads to small amplitude errors (see Figure 4.8) that stay bounded throughout the computation.

4.4.2 1D Sod shock tube

Sod shock tube is a standard Riemann problem for Euler's equations (Toro (2009)). In the case of two-phase flows, a similar test is presented in Chanteperdrix, Villedieu, and Vila (2002) for the 3-equation system. Here, we perform two types of shock tests. In the first one, the shock occurs in a gas-liquid mixture ($\alpha = 0.5$ everywhere). A rarefaction and a shock waves are created and propagate in the mixture. In the second test, a two-phase interface separates a gas phase ($\alpha = 1.0 - 1.0 \cdot 10^{-8}$) from a liquid phase ($\alpha = 1.0 \cdot 10^{-8}$). In the gas phase, pressure is higher than in the liquid phase. A shock occurs at the

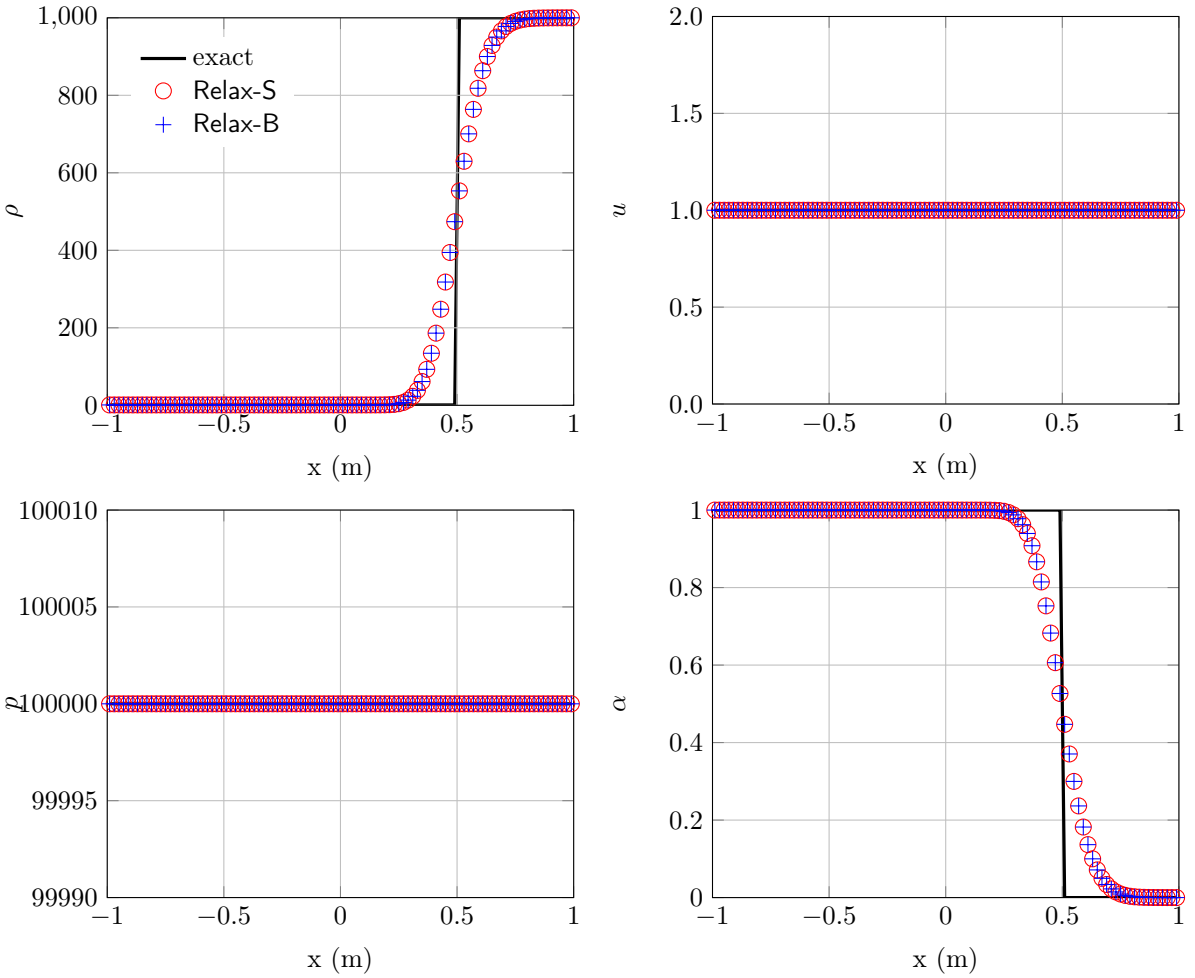


Figure 4.4: *3-equation system: Relax-S (red circles) and Relax-B (blue crosses) solutions on $N = 100$ cells and exact solution for the advection test at time $t = 5.0 \cdot 10^{-1}$ s.*

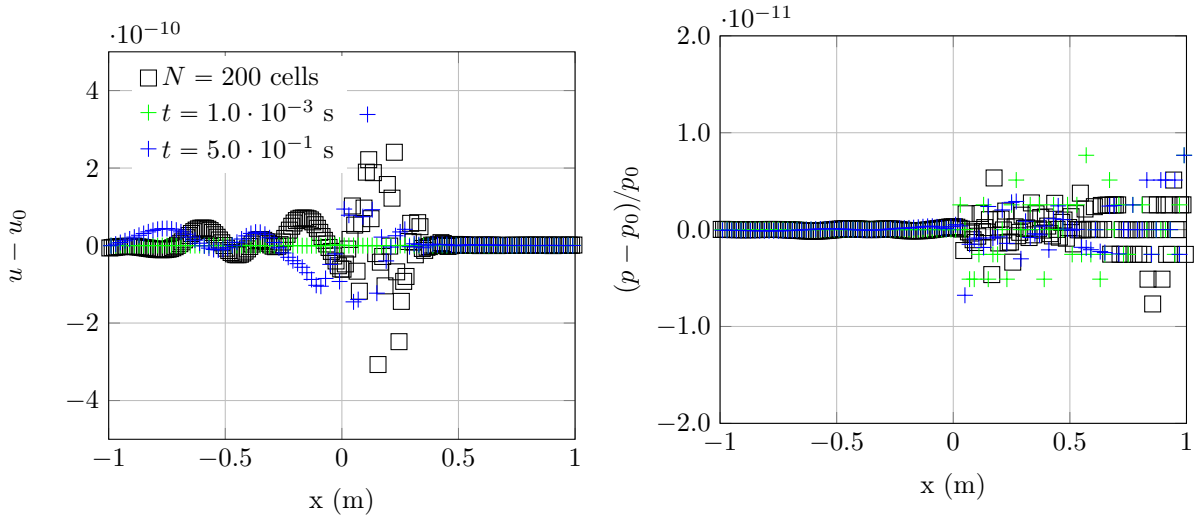


Figure 4.5: *Relax-B solutions for the 3-equation system and the advection test: at time $t = 1.0 \cdot 10^{-3}$ s for $N = 100$ cells (green crosses); at time $t = 5.0 \cdot 10^{-1}$ s for $N = 100$ (blue crosses) and for $N = 200$ cells (black squares).*

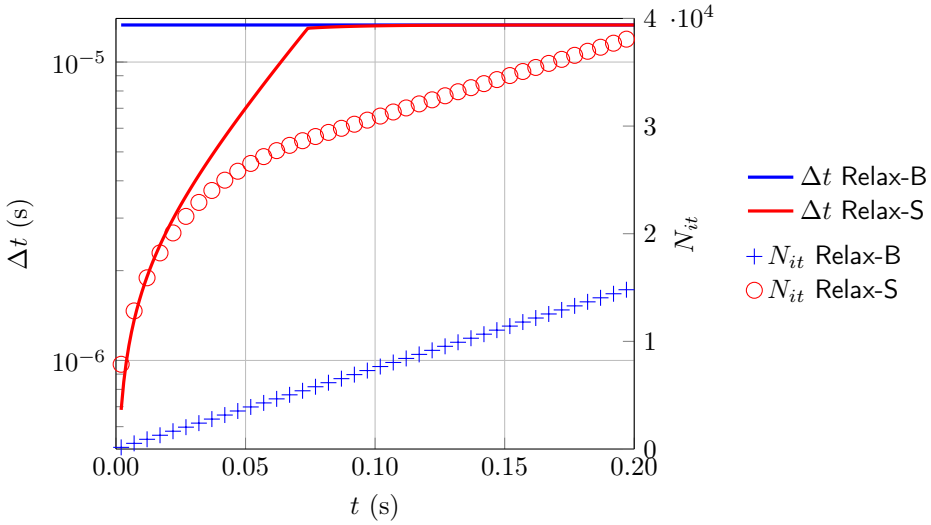


Figure 4.6: *3-equation system: comparison of the values of Δt between Relax-S and Relax-B (solid lines) and comparison of the number of time iterations performed up to the simulation time t (markers) for the advection test case, $\Delta x = 0.02$ m.*

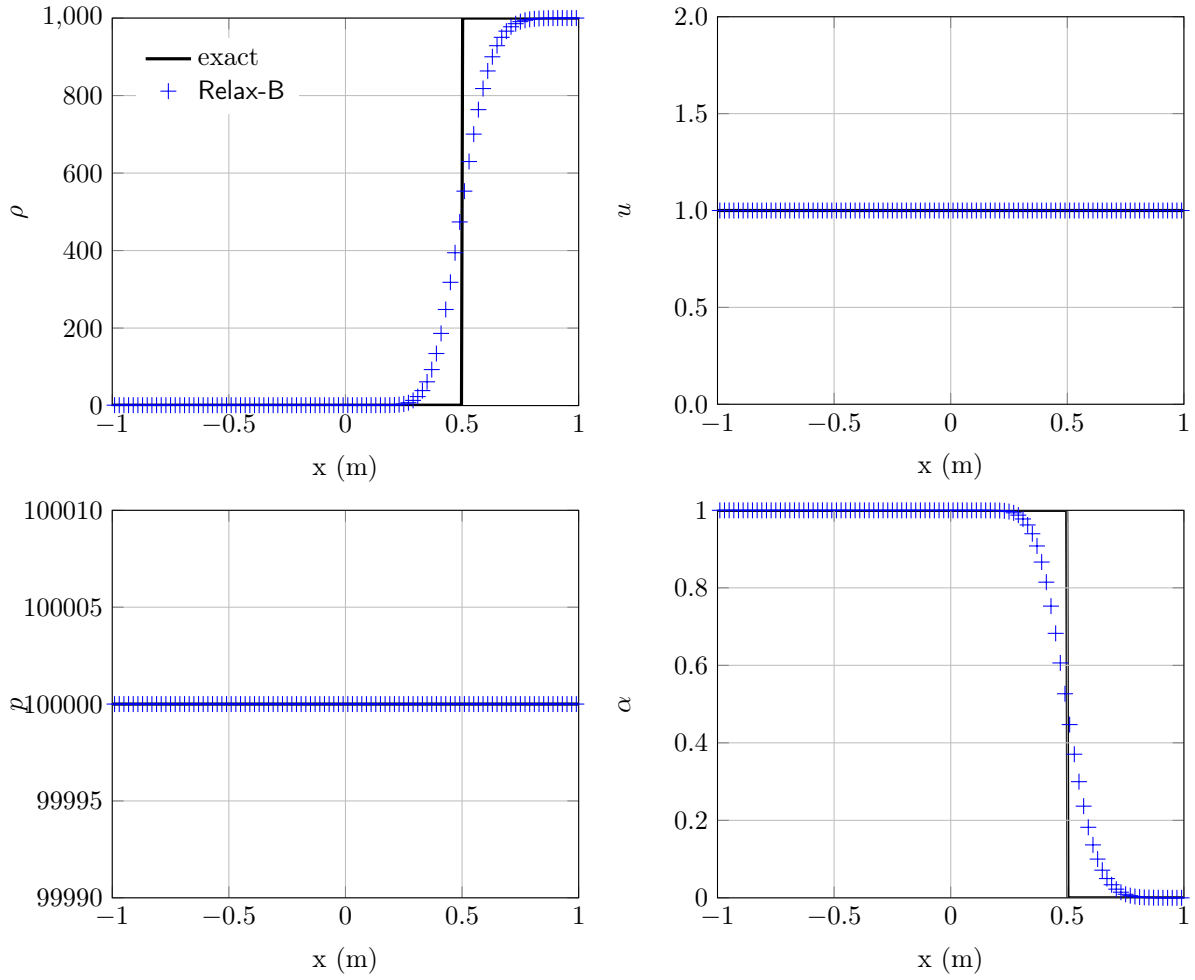


Figure 4.7: *4-equation system: Relax-B solution on $N = 100$ cells and exact solution for the advection test at time $t = 5.0 \cdot 10^{-1}$ s.*

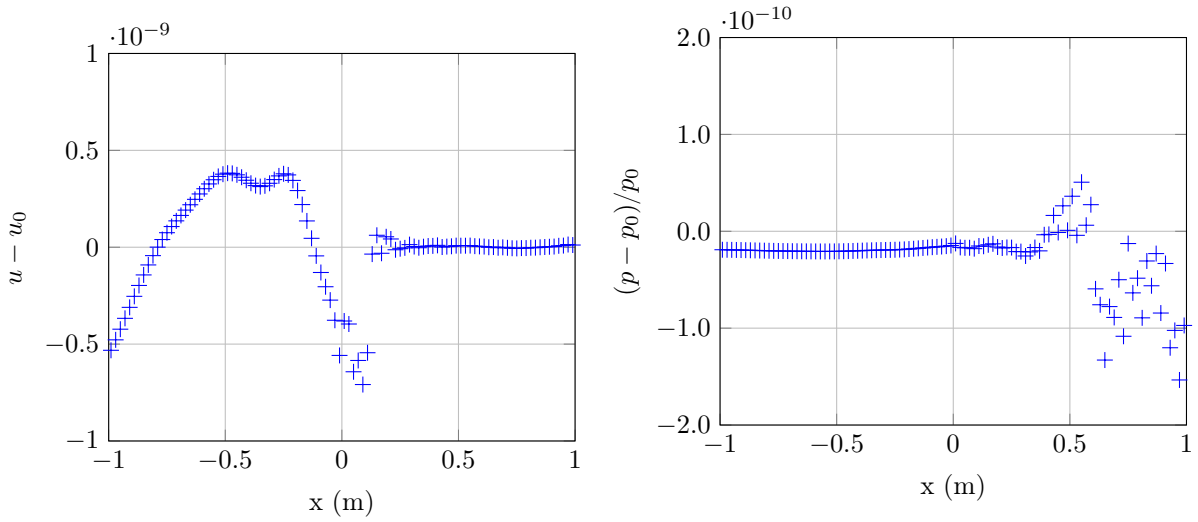


Figure 4.8: *4-equation system: Relax-B solution on 100 cells and exact solution for the advection test at time $t = 5.0 \cdot 10^{-1}$ s. Closer view on velocity and pressure.*

interface, and generates a rarefaction wave in the gas and a shock wave in the liquid. This test is also studied in Chanteperdrix, Villedieu, and Vila (2002).

Mixture shock tube

The initial conditions for this test are given in the table below:

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$p_L = 5.0 \cdot 10^5$	$p_R = 1.0 \cdot 10^5$
mass fraction	$Y_0 = 4.4 \cdot 10^{-3}$	$Y_0 = 1.0 \cdot 10^{-3}$
velocity	$u_0 = 0.0$	$u_0 = 0.0$
volume fraction	$\alpha_0 = 0.5$	$\alpha_0 = 0.5$
vibration	$w_0 = 0.0$	$w_0 = 0.0$

In Figure 4.9, we compare the solution of the 3-equation system solved with Relax-B with the exact solution (see Appendix A). We can see that numerical diffusion occurs mainly near the contact discontinuity, while shock is well-resolved.

In Figure 4.10 are presented the comparisons between Relax-B scheme and the exact solution for the 4-equation system. There, the final time of the simulation is shorter, as the characteristic waves propagate with a larger velocity. This is also why we cannot observe much diffusion at the contact discontinuity yet. Another difference with the solution of the 3-equation system is the profile of α: in the 3-equation case, this variable is computed using ρ and Y, while in the case of the 4-equation system, it is a transported variable. Since the initial profile of α is uniform, it stays uniform in the case of the 4-equation system.

Now let us study whether one can recover the solution of the 3-equation system by projecting the solution of the 4-equation system on the mechanical equilibrium manifold, following the procedure of Chanteperdrix, Villedieu, and Vila (2002). The results for the mixture shock tube are presented in Figure 4.11. One can see that with this method, numerical diffusion is dramatically high, and the solution of the 3-equation system is poorly approximated. We will discuss this issue in Chapter 5 and propose new ideas for a numerical cure.

Two-phase shock tube

The initial conditions are given by:

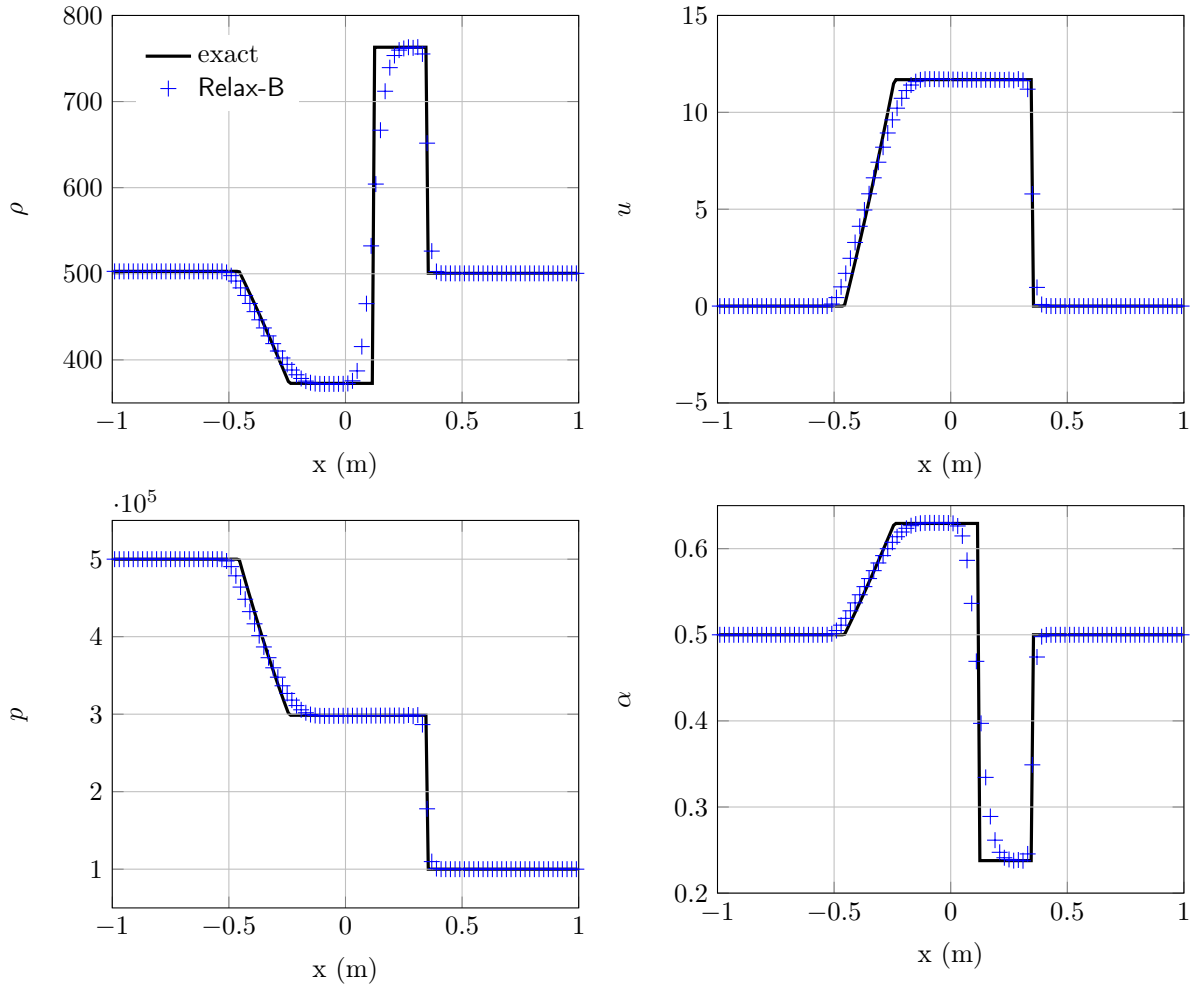


Figure 4.9: *3-equation system: Relax-B solution on 100 cells and exact solution for the mixture shock test at time* $t = 1.0 \cdot 10^{-2}$ *s.*

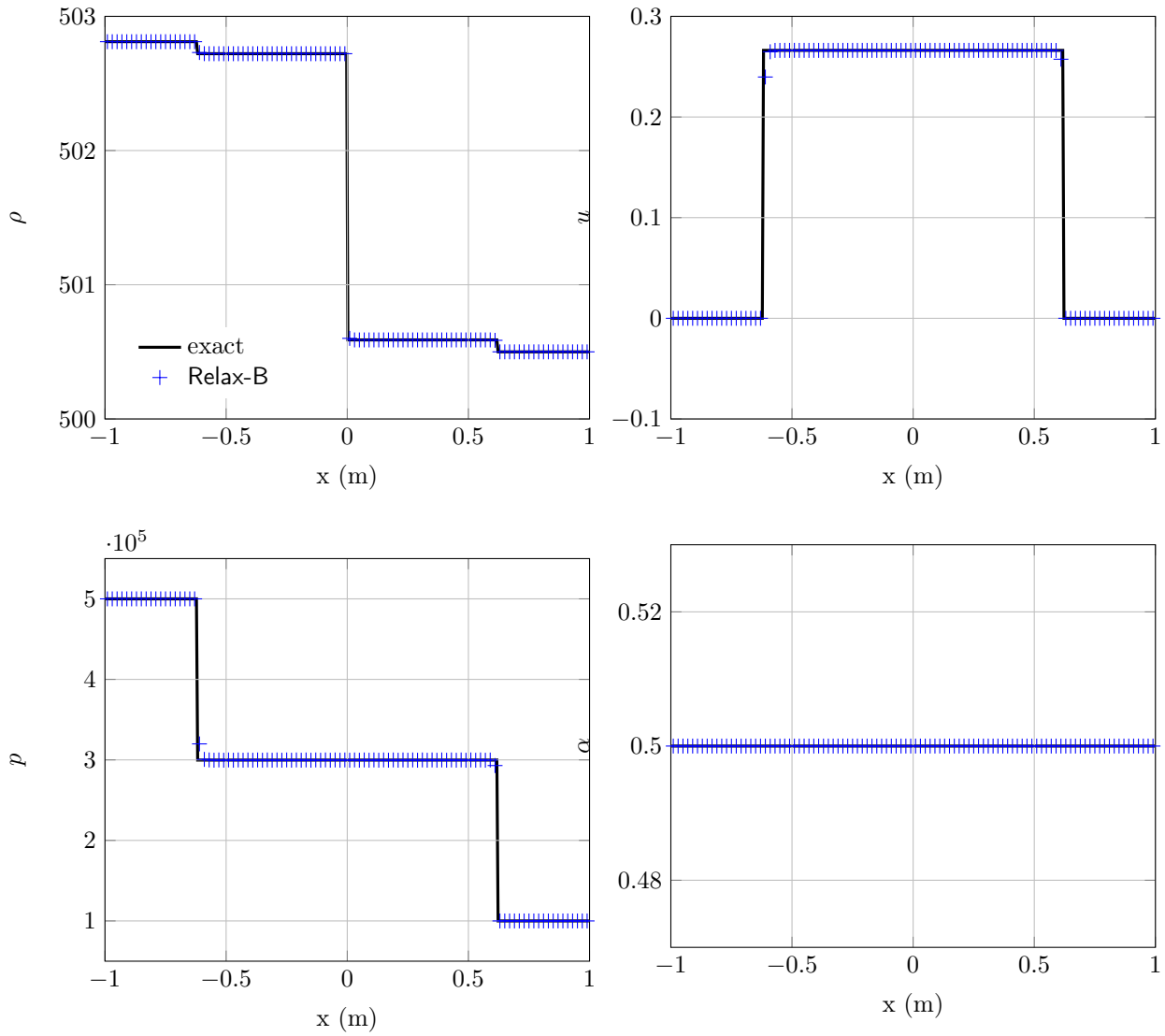


Figure 4.10: *4-equation system: Relax-B solution on 100 cells and exact solution for the mixture shock test at time* $t = 4.0 \cdot 10^{-4}$ *s.*

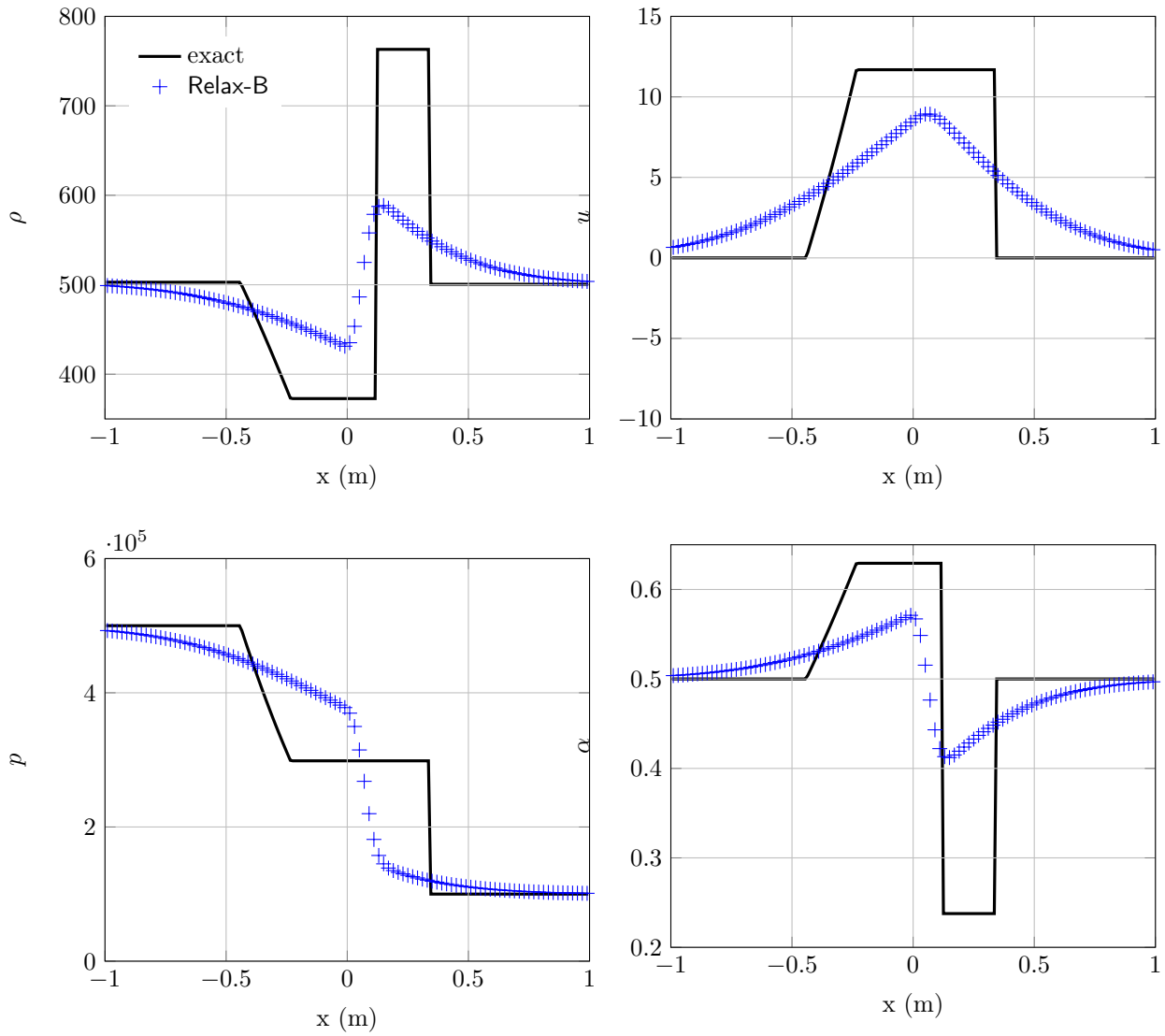


Figure 4.11: *4-equation system and relaxation to equilibrium: Relax-B solution on 100 cells and exact solution of the 3-equation system for the mixture shock test at time* $t = 1.0 \cdot 10^{-2}$ *s.*

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$p_L = 1.1 \cdot 10^5$	$p_R = 1.0 \cdot 10^5$
mass fraction	$Y_L = 1.0 - 1.0 \cdot 10^{-8}$	$Y_R = 1.0 \cdot 10^{-11}$
velocity	$u_0 = 0.0$	$u_0 = 0.0$
volume fraction	$\alpha_L = 1.0 - 1.0 \cdot 10^{-8}$	$\alpha_R = 1.0 \cdot 10^{-8}$
vibration	$w_0 = 0.0$	$w_0 = 0.0$

For this test only, we change the values of the phasic sound velocities: they are now given by $c_1 = 3.4$ m/s and $c_2 = 15.0$ m/s. The aim of this change is to reduce the computational cost of the test by decreasing the CFL constraint, and consequently increasing the time step Δt, in order to recover the configuration of Chanteperdrix, Villedieu, and Vila (2002).

Results and comparisons with the exact solutions are given for the 3-equation and 4-equation systems in Figure 4.12 and Figure 4.13 respectively. For this test, one can note that the exact solutions for the two systems of equations are close, so are the numerical solutions. This is due to the fact that the mixture zone, where $0 < \alpha < 1$, is reduced to a thin zone near the contact discontinuity. In the rest of the domain, the configurations are close to pure phases. In these zones, the physics described by the 3-equation and 4-equation systems is nearly the same. Thus, for this test, the procedure consisting in projecting the values of α and p on their equilibrium value may give good results.

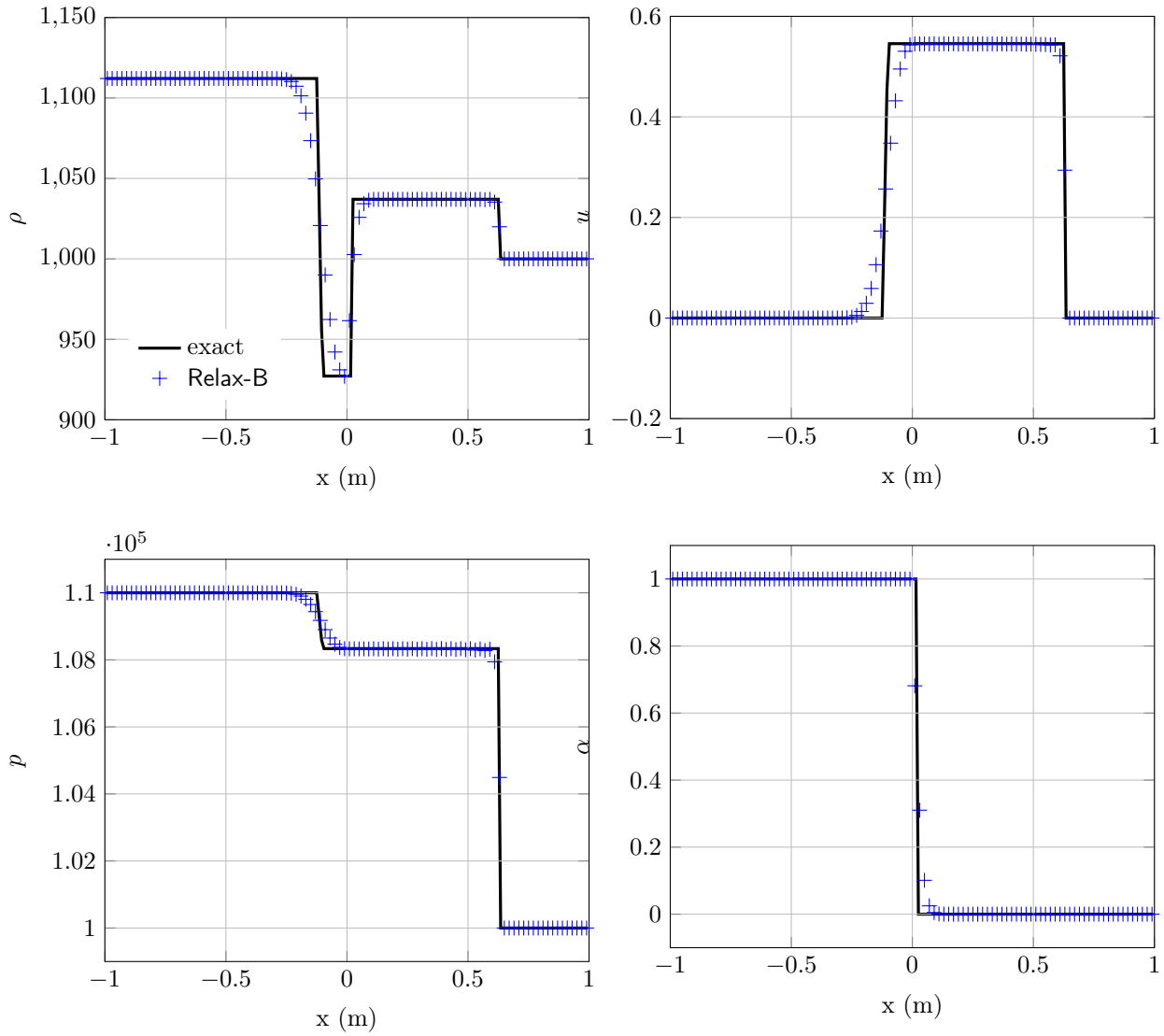


Figure 4.12: *3-equation system: Relax-B solution on 100 cells and exact solution for the shock test with gas-liquid interface at time* $t = 4.0 \cdot 10^{-2}$ *s.*

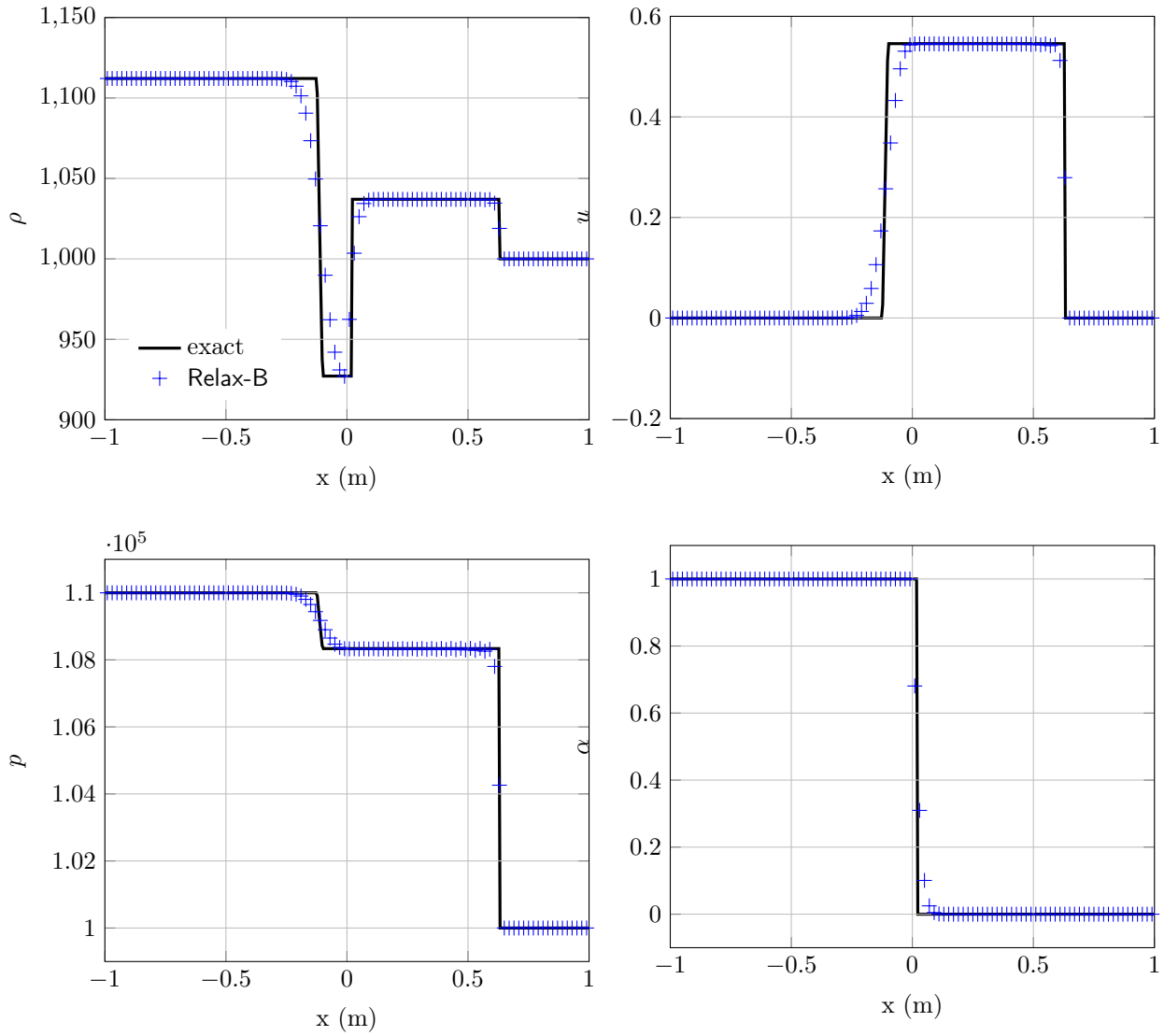


Figure 4.13: *4-equation system: Relax-B solution on 100 cells and exact solution for the shock test with gas-liquid interface at time* $t = 4.0 \cdot 10^{-2}$ *s.*

4.5 Conclusion

In this chapter, we have proposed a finite volume numerical scheme to solve the convective part of the two-phase systems of equations. We have first considered the Relax-S scheme, that is used in several other works from the literature (Girardin (2014) for example). We have shown that, in fact, this scheme presents some shortcomings when two-phase interfaces are considered: in this configuration, the velocities of the waves are largely overestimated and the CFL condition on the time step is extremely constraining. This is why, we have proposed a second scheme, Relax-B, inspired from Bouchut (2004). This new scheme still necessitates some developments and study (so as to ensure its entropic character), but in practice, the numerical tests that have been performed have shown good results. The global numerical scheme is also made of a specific discretization for the volume fraction α (and also for w) for the 4-equation system, as is proposed in Kokh and Lagoutière (2010). This discretization enables to preserve uniform pressure and velocity profiles, which is a condition to limit the creation of pressure instabilities (Abgrall (1996)). The first series of numerical tests, presented at the end of the chapter, show that the Relax-B scheme provides robust results and is clearly more advantageous compared to Relax-S scheme. We have also observed that in the mixture zones, the physics described by the 4-equation system is different from the physics solved by the 3-equation system and in these zones, a strategy consisting in solving the 4-equation system and then projecting on the equilibrium values for α and p does not allow to recover an accurate approximation of the 3-equation solution. This last aspect is further discussed in Chapter 5, where the source term effects are integrated and tests with a physical meaning are considered.

Chapter 5

Source term integration and numerical dispersion relations

Let us recall that the systems which have been derived in Chapter 1 can be written in the form of balance laws:

$$\partial_t \mathbf{W} + \operatorname{div}(\mathbf{F}(\mathbf{W})) = \mathbf{S}(\mathbf{W}), \tag{5.1}$$

where $\mathbf{W}$ are the conservative variables, $\mathbf{F}$ are the physical fluxes and $\mathbf{S}$ are the source terms. Using an operator splitting, as presented in the Introduction of Part II, one can solve the different parts of (5.1) in successive steps. First, one solves the convective part of the equations:

$$\partial_t \mathbf{W} + \operatorname{div}(\mathbf{F}(\mathbf{W})) = 0,$$

and the source terms are integrated next:

$$\partial_t \mathbf{W} = \mathbf{S}(\mathbf{W}). \tag{5.2}$$

We have studied approximate Riemann solvers in Chapter 4 that will be used to solve the convective part. Let us look at the integration of the source terms in this chapter. The first natural approach is to consider classic Ordinary Differential Equations (ODE) solvers, that are proposed by the many works of the literature (mentioned in the introduction of Part II, and referenced in Hairer and Wanner (1996)). However, for the integration of the source term of the 5-equation system, these solvers seem to fail in stiff configurations. This is the case of Radau5 (Hairer and Wanner (1996)) used to integrate the source term of the 5-equation system in Appendix B. In practice, we do not want to deal with possible errors raised from ODE solvers. Moreover, in a first approach, we are looking for quite simple solutions for the source terms integration, at a *reasonable* accuracy and a *reasonable* computational cost. These two points justify the work presented in this chapter: we study approximate solutions to equation (5.2) in the specific case of the two-phase models of Chapter 1, so as to propose a robust strategy of integration even in the stiffest cases.

The second issue which is treated in this chapter concerns the asymptotic relations between the different models. Indeed, we have seen at the end of Chapter 2 that the dispersion relations of the models are related in two ways:

- by the frequency regime of an acoustic wave,
- by the values of the parameters ϵ and ν of the source terms.

Numerically, we would like to recover these relations. For example, this could be useful to switch from one system of equations to another according to given threshold values of the parameters, and then reduce the computational cost. In the second part of this chapter, we test the whole numerical strategy, made of the Lie splitting, an approximate Riemann solver and and ODE solver. Our aim is to recover the analytical dispersion relations numerically. In this part, we will also raise a difficulty which is related to a too large numerical diffusion in the asymptotic regimes.

5.1 Integration of the 4-equation relaxation source term

In this section, we are interested in the integration of the source term of the 4-equation system. Let us recall that this source term appears in the equation on volume fraction α, which writes:

$$\partial_t \alpha + \mathbf{u} \cdot \boldsymbol{\nabla} \alpha = \frac{p_1(\rho, Y, \alpha) - p_2(\rho, Y, \alpha)}{\epsilon}.$$

Let us also recall that the pressure laws are barotropic, in the form:

$$\rho_k \mapsto p_k(\rho_k), \quad k = 1, 2,$$

where $\rho_1 = (\rho Y)/\alpha$, $\rho_2 = (\rho(1-Y))/(1-\alpha)$ and the phasic sound velocities c_k are defined by

$$p_k'(\rho_k) = c_k^2, \quad k = 1, 2.$$

From the Lie splitting procedure, taking into account the source term in the numerical resolution of the 4-equation system comes down to solve an ODE in time, that reads:

$$\mathrm{d}_t \alpha = \frac{p_1(\overline{\rho}, \overline{Y}, \alpha) - p_2(\overline{\rho}, \overline{Y}, \alpha)}{\epsilon},$$

where the variables ρ and Y are kept constant during this time integration and are thus noted with a $\overline{\cdot}$ sign. Let us define the function $F(\alpha)$ as:

$$F : \begin{pmatrix}]0,1[& \to & \mathbb{R} \\ \alpha & \mapsto & p_1(\overline{\rho}, \overline{Y}, \alpha) - p_2(\overline{\rho}, \overline{Y}, \alpha) \end{pmatrix}.$$

In a first time, let us study the properties of the following ODE:

$$\mathrm{d}_t \alpha = \frac{F(\alpha)}{\epsilon}. \tag{5.3}$$

5.1.1 Stationary case

First, let us study the case when the initial condition for ODE (5.3) is at equilibrium, meaning $\alpha(0) = \alpha_{eq}$. Then, by definition, $p_1(\overline{\rho}, \overline{Y}, \alpha_{eq}) = p_2(\overline{\rho}, \overline{Y}, \alpha_{eq})$ and then: $F(\alpha_{eq}) = 0$. The integration is then trivial, and the solution is $\alpha(t) = \alpha_{eq}$.

α_{eq} is the unique stationary value within $]0,1[$, as by definition, it is the unique value in $[0,1]$ such that $p_1(\overline{\rho}, \overline{Y}, \alpha_{eq}) = p_2(\overline{\rho}, \overline{Y}, \alpha_{eq})$. As a consequence, if $\alpha(0) \neq \alpha_{eq}$, then $\alpha(t) \neq \alpha_{eq}$ $\forall t > 0$.

5.1.2 Study of the differential equation and exact solution in the non-stationary case

When $\alpha(0) \neq \alpha_{eq}$, let us solve the ODE (5.3) exactly. Let us write it in the form:

$$E(\alpha)\, \mathrm{d}_t \alpha = \frac{1}{\epsilon},$$

where the function E is defined as:

$$E : \begin{pmatrix}]0, \alpha_{eq}[\bigcup]\alpha_{eq}, 1[& \to & \mathbb{R} \\ \alpha & \mapsto & \dfrac{1}{F(\alpha)} \end{pmatrix}.$$

In the following, we assume that $\overline{Y} \in]0,1[$ and $\overline{\rho} > 0$. In the cases $\overline{Y} = 0$ and $\overline{Y} = 1$, F is not well defined. Indeed, when the mass fraction Y is equal to 0 or 1, volume fraction α is equal to 0 or 1 in the same way. Then, one of the densities ρ_k, $k = 1, 2$ is equal to the total density ρ while the other is not defined (because of a division of 0 by 0). For the rest of the present study, we will then exclude the cases $\overline{Y} = 0$ or $\overline{Y} = 1$, and from a numerical point of view, when these configurations are encountered, we will assume that $\mathrm{d}_t \alpha = 0$.

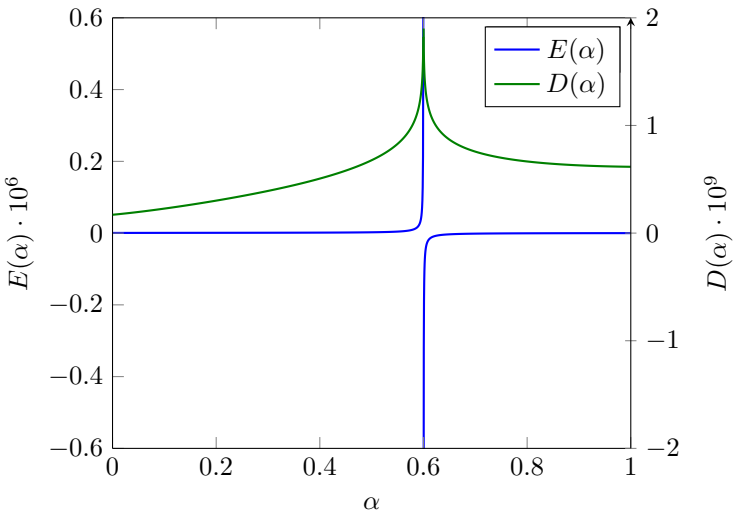


Figure 5.1: *Illustration of functions E and D in the specific case of linearized pressure laws (5.5), with $\alpha_{eq} = 0.6$ and $\alpha_- = -0.1$.*

5.1.2.1 General form of the solution

Here, the point is to develop a general formulation of the solution, whatever the choice of the barotropic pressure laws.

Properties of $E(\alpha)$:

Let us study the function $E(\alpha)$.

Proposition 1. Considering barotropic pressure laws, function E has the following properties:

1. $E(\alpha)$ is increasing over the interval $]0, \alpha_{eq}[$ and is increasing over the interval $]\alpha_{eq}, 1[$,
2. $E(\alpha) \underset{\alpha \to 0^+}{\longrightarrow} 0$ and $E(\alpha) \underset{\alpha \to 1^-}{\longrightarrow} 0$,
3. $E(\alpha) \underset{\alpha \to \alpha_{eq}^-}{\longrightarrow} +\infty$ and $E(\alpha) \underset{\alpha \to \alpha_{eq}^+}{\longrightarrow} -\infty$,
4. $E(\alpha) \geqslant 0$ for $\alpha \in]0, \alpha_{eq}[$ and $E(\alpha) \leqslant 0$ for $\alpha \in]\alpha_{eq}, 1[$.

Remark 20. *the fourth property is a direct consequence of the two first properties.*

Proof of Proposition 1. To prove the form of E, let us study F. According to the barotropic pressure laws, one has:

$$F'(\alpha) = -c_1^2 \frac{\overline{\rho}\overline{Y}}{\alpha^2} - c_2^2 \frac{\overline{\rho}(1-\overline{Y})}{(1-\alpha)^2}.$$

It is then obvious that $F'(\alpha) \leqslant 0$ when $\alpha \in]0, 1[$. F is then decreasing over $]0, 1[$ and cancels at $\alpha = \alpha_{eq}$. Moreover, $F(\alpha) > 0$ over $]0, \alpha_{eq}[$ and $F(\alpha) < 0$ over $]\alpha_{eq}, 1[$. One can easily show also that $F(\alpha) \underset{\alpha \to 0^+}{\longrightarrow} +\infty$ and $F(\alpha) \underset{\alpha \to 1^-}{\longrightarrow} -\infty$. The properties for E are directly derived by applying the inverse of F on each subdomain $]0, \alpha_{eq}[$ and $]\alpha_{eq}, 1[$. □

E is represented in Figure 5.1 as a function of α for linearized pressure laws (see below).

Now let us define the function $D(\alpha)$ as one prime function of $E(\alpha)$ over $]0, \alpha_{eq}[\bigcup]\alpha_{eq}, 1[$. According to Proposition 1, the properties of D are given by

Proposition 2. 1. $D(\alpha)$ is increasing on$]0, \alpha_{eq}[$ and is decreasing on $]\alpha_{eq}, 1[$

2. $D(\alpha) \underset{\alpha \to \alpha_{eq}^-}{\longrightarrow} +\infty$ and $D(\alpha) \underset{\alpha \to \alpha_{eq}^+}{\longrightarrow} +\infty$.

Proof of Proposition 2. The first property is a consequence of property 4. of Proposition 1. The second property is derived from the property 3. of Proposition 1. □

Solution of the ODE:

The integration of the ODE (5.3) from $t = 0$ to $t = t_f > 0$ gives:

$$D(\alpha(t_f)) - D(\alpha(0)) = \frac{t_f}{\epsilon}. \tag{5.4}$$

In (5.4), the solution $\alpha(t_f)$ is given implicitly. There may not be any analytical formulation for it, as we will see for the linearized pressure laws (an illustration of D is given in Figure 5.1 in that case). However, we have the following properties:

Proposition 3. If $\alpha(0) \in]0, \alpha_{eq}[$, then there exists a unique $\alpha(t_f) \in]0, \alpha_{eq}[$ satisfying equation (5.4). Similarly, if $\alpha(0) \in]\alpha_{eq}, 1[$, then there exists a unique $\alpha(t_f) \in]\alpha_{eq}, 1[$ satisfying equation (5.4).

Proof of Proposition 3. From Proposition 2, we know that D is increasing over $]0, \alpha_{eq}[$, and $D(\alpha) \underset{\alpha \to \alpha_{eq}^-}{\longrightarrow} +\infty$.

If $\alpha(0) \in]0, \alpha_{eq}[$, as $t_f > 0$, then $D(\alpha(t_f)) > D(\alpha(0))$ and according to the variations of D on this interval, $\alpha(t_f)$ exists and is unique.

The case $\alpha(0) \in]\alpha_{eq}, 1[$ is treated the same way. □

Proposition 4. If $\alpha(0) \in]0, \alpha_{eq}[\bigcup]\alpha_{eq}, 1[$, then $|\alpha(t_f) - \alpha_{eq}| < |\alpha(0) - \alpha_{eq}|$. This means that the solution is getting closer to the equilibrium state.

Proof of Proposition 4. By assuming that the integration of ODE (5.3) is continuous, if $\alpha(0) \in]0, \alpha_{eq}[$, then according to Proposition 3, $\alpha(t_f)$ exists and is unique in $]0, \alpha_{eq}[$. Since D is increasing on this interval, $D(\alpha(t_f)) > D(\alpha(0))$ and $\alpha(t_f) > \alpha(0)$. Moreover $\alpha(t_f) < \alpha_{eq}$. Thus we have

$$0 < \alpha_{eq} - \alpha(t_f) < \alpha_{eq} - \alpha(0).$$

We can apply the same reasoning for the interval $\alpha(0) \in]\alpha_{eq}, 1[$ by using the decreasing character of D, and get then the inequality:

$$0 < \alpha(t_f) - \alpha_{eq} < \alpha(0) - \alpha_{eq}.$$

Combining these two inequalities proves Proposition 4. □

From Proposition 3 and Proposition 4, we know that the solution of the ODE, when not initially at equilibrium, strictly increases (or decreases) towards a stationary solution, that does not lie on the other side of α_{eq}. Now, α_{eq} is the unique stationary value in $]0, 1[$. Then, in long times ($t_f \to \infty$), the solution tends to α_{eq}.

Remark 21. *The properties of F can also be deduced from the properties of the entropy. Indeed, let us assume that the entropy η of the 4-equation system is given by:*

$$\eta(\rho, Y, u, \alpha) = \frac{1}{2}\rho u^2 + f(\rho, Y, \alpha),$$

where f is Helmholtz free energy, which is a convex function of the variables (see proof in Serre (2007)). Then since

$$F(\alpha) = -\rho\, \partial_\alpha \eta|_{\rho, Y, u}\,,$$

$F'(\alpha)$ is negative and F is a decreasing function over $]0, 1[$.

5.1.2.2 $E(\alpha)$ and $D(\alpha)$ for linearized pressure laws:

Now let us consider linearized pressure laws in the form:

$$p_k = p^o + c_k^2 \left(\rho_k(\rho, Y, \alpha) - \rho_k^o\right), \quad k = 1, 2. \tag{5.5}$$

With these pressure laws, function E writes:

$$E(\alpha) = -\frac{\alpha(1-\alpha)}{q_0\alpha^2 + (q - q_0)\alpha - \overline{\rho}\overline{Y}c_1^2},$$

with $q_0 = \rho_2^o c_2^2 - \rho_1^o c_1^2$ and $q = \overline{\rho}\overline{Y}c_1^2 + \overline{\rho}(1-\overline{Y})c_2^2$. The second order polynomial in the function denominator always has two real roots, α_{eq} and α_-. We know that $\alpha_{eq} \in]0,1[$ (if $Y \in]0,1[$). Moreover one can show that $\alpha_- < 0$, since the product of the two roots must satisfy $\alpha_{eq}\alpha_- = -\overline{\rho}\overline{Y}c_1^2/q_0$ and since $q_0 > 0$ for the state laws that are chosen such that $\rho_2^o c_2^2 > \rho_1^o c_1^2$ (in the other case, we can switch the indices, and consider $\tilde{\alpha} = 1 - \alpha$).

The decomposition of E into partial fractions reads:

$$E(\alpha) = B_0 + \frac{B_1}{\alpha - \alpha_{eq}} + \frac{B_2}{\alpha - \alpha_-} \tag{5.6}$$

with:

$$B_0 = \frac{1}{q_0},$$
$$B_1 = \frac{\alpha_{eq}(1 - \alpha_{eq} - \alpha_-)}{q_0(\alpha_- - \alpha_{eq})},$$
$$B_2 = \frac{\alpha_-(1 - \alpha_{eq} - \alpha_-)}{q_0(\alpha_{eq} - \alpha_-)}.$$

Finally, it is easy to integrate (5.6) to obtain:

$$D(\alpha) = B_0\alpha + B_1 \ln(|\alpha - \alpha_{eq}|) + B_2 \ln(\alpha - \alpha_-), \quad \forall \alpha \in]0, \alpha_0[\bigcup]\alpha_{eq}, 1[. \tag{5.7}$$

Illustrations of functions E and D are given in Figure 5.1.

We do not know the inverse function of D on each subinterval $]0, \alpha_{eq}[$ and $]\alpha_{eq}, 1[$. However, due to its properties (Proposition 2) and the properties of its derivative E, classic root-finding algorithms can be applied.

Remark 22. *We can note that the variations of $D(\alpha)$ in $]0, \alpha_{eq}[$ and $]\alpha_{eq}, 1[$ are larger than the variations of α, since $D(\alpha)$ is made of logarithmic terms applied to values inferior to 1. An approximate solution $\alpha(t_f)$ to (5.4) can then be given with a high accuracy, while the equation defined by (5.4) is satisfied with a lower accuracy.*

5.1.2.3 An approximate solution to the ODE:

To complete this section, let us consider an approximate solution to the ODE (5.3), obtained by assuming that $\alpha(t = 0)$ and $\alpha(t = t_f)$ are close enough for a Taylor expansion in the vicinity of $\alpha(t = 0)$ to be valid:

$$F(\alpha(t_f)) = F(\alpha(0)) + (\alpha(t_f) - \alpha(0))\, F'(\alpha(0)) + O\left(\alpha(t_f) - \alpha(0)\right)^2.$$

Or more generally, for a reference value α^* and a value of α close to α^*, we have:

$$F(\alpha) = F(\alpha^*) + (\alpha - \alpha^*) F'(\alpha^*) + O\left(\alpha - \alpha^*\right)^2, \tag{5.8}$$

and then, at first order the ODE (5.3) becomes:

$$\mathrm{d}_t\alpha = \frac{1}{\epsilon}\left(F(\alpha^*) + (\alpha - \alpha^*) F'(\alpha^*)\right),$$

whose analytical solution is given by

$$\alpha(t_f) = \alpha(0)\exp\left(\frac{F'(\alpha^*)t_f}{\epsilon}\right) + \frac{F(\alpha^*) - \alpha^* F'(\alpha^*)}{F'(\alpha^*)}\left(\exp\left(\frac{F'(\alpha^*)t_f}{\epsilon}\right) - 1\right). \tag{5.9}$$

For the linearized pressure laws, one will moreover have:

$$F(\alpha) = c_1^2\frac{\overline{\rho}\overline{Y}}{\alpha} - c_2^2\frac{\overline{\rho}(1-\overline{Y})}{1-\alpha} - c_1^2\rho_1^o + c_2^2\rho_2^o,$$
$$F'(\alpha) = -c_1^2\frac{\overline{\rho}\overline{Y}}{\alpha^2} - c_2^2\frac{\overline{\rho}(1-\overline{Y})}{(1-\alpha)^2}.$$

5.1.3 Solving the 4-equation source term numerically

Now, we are interested in accurate, robust and not too costly approximate solutions of (5.3). Four strategies can be considered:

- a strategy consisting in using the exact solution of (5.3), given by (5.4) and which necessitates a root-finding algorithm;
- a coarse and *naive strategy*, based on the integration of the approximate ODE from time $t = 0$ to time $t = t_f$:

$$\mathrm{d}_t\alpha = \frac{F(\alpha(t=0))}{\epsilon};$$

- implicit integrations that consist in solving, for instance the first order implicit Euler:

$$\mathrm{d}_t\alpha = \frac{F(\alpha(t=t_f))}{\epsilon}.$$

- strategies based on the exact solution of the approximate ODE (5.9).

Higher-order methods, which were mentioned in the introduction of Part II have not been considered here, since we are comparing quite rough integration methods, but this could be done in a future work.

In the following, we suppose that the integration of the ODE is performed between time $t = t^n$ and time $t = t^{n+1} = t^n + \Delta t$. We note $\alpha^n = \alpha(t^n)$ and $\alpha^{n+1} = \alpha(t^{n+1})$ the solution obtained with the different strategies after **one time iteration** of the different methods. Again, this choice is made for a first comparison between rough integration methods. In a final strategy, and because of the operator splitting method, a more accurate integration could be envisioned, involving multiple time iterations.

We recall that ρ and Y are constant during the source term integration.

5.1.3.1 Exact solution approximation

This approach consists in using the relation (5.4) to estimate α^{n+1}. In the case of linearized pressure laws, the function D is given by (5.7), but its inverse function is not known. Standard root-finding algorithms can be used due to the properties of D and its derivative E presented above. This method based on the exact solution, will be used in the following comparisons as a reference solution.

5.1.3.2 Coarse integration

Here, let us present an explicit strategy which is very coarse and is not A-stable. We try however to improve it. Let us consider the approximate explicit source term integration, which consists in solving:

$$\mathrm{d}_t\alpha = \frac{F(\alpha^n)}{\epsilon}.$$

The solution for one time iteration Δt is given by:

$$\alpha^{n+1} = \alpha^n + \Delta t \frac{F(\alpha^n)}{\epsilon}. \tag{5.10}$$

One can see that, according to the values of Δt and ϵ, the second term of the right-hand side of (5.10) may be very large. In our numerical strategy, Δt is computed according to a CFL condition based on the conservative part of the system. This computation is thus independent of the value of ϵ. However, the value of α should not leave the interval $[0, 1]$, nor even overpass (or underpass) the value α_{eq} if $\alpha^n < \alpha_{eq}$ (or $\alpha^n > \alpha_{eq}$ respectively). One must then adapt the time step Δt before using (5.10).

A first option to ensure that $\alpha^{n+1} \in [0, \alpha_{eq}[$ (or $\alpha^{n+1} \in]\alpha_{eq}, 1]$ according to the initial condition) would be to adapt the time step to this condition, or divide the explicit integration in multiple time substeps. But this option is not interesting because one can see that if ϵ is arbitrarily small, one has to strongly reduce Δt or make an arbitrarily large number of substeps, which can be computationally expensive.

Let us instead consider that, from the CFL estimation of Δt:

$$\tau_{\text{relax}} \ll \Delta t \quad \Rightarrow \quad \alpha^{n+1} = \alpha_{eq}, \tag{5.11}$$

where τ_{relax} is a characteristic relaxation time depending on ϵ, ρ^n and Y^n. The question is now to be able to correctly estimate τ_{relax}, or to be able to set $\alpha^{n+1} = \alpha_{eq}$ when this option seems correct.

According to the previous section 5.1.2.1, one knows that α tends to α_{eq} and that it does not cross the value α_{eq}. Then, let us set a strategy for the source term integration as:

1. compute α^{n+1} using the formula (5.10) and compute the equilibrium value α_{eq} using ρ^n and Y^n,
2. compare α^n with α_{eq} and α^{n+1} with α_{eq}:
   - if $(\alpha^n - \alpha_{eq}) \cdot (\alpha^{n+1} - \alpha_{eq}) < 0$, this means that during the integration, α has overpassed the equilibrium state, which cannot happen physically. Then we enforce $\alpha^{n+1} = \alpha_{eq}$,
   - else, this means that $\Delta t < \tau_{\text{relax}}$ and there is no problem with the explicit integration.

This very coarse explicit integration assumes that the pressure difference remains constant during the temporal integration. Hence it is less accurate than the approximate integrations presented in the following.

5.1.3.3 A first approximate relaxation

Due to the form of the source term, which is non-linear, we know that the relaxation is not performed in an exact exponential way. However, in an approximate approach, we could wonder if such an exponential integration is far from the real solution, and, as we have seen with the approximate solutions of paragraph 5.1.2.3, for which the exponential approximation appears naturally from the Taylor expansion of the source term.

Let us then suppose that we are solving a linear ODE of the form:

$$\mathrm{d}_t\alpha = \frac{\alpha_{eq} - \alpha}{\beta},$$

where the parameter β is given by:

$$\beta = (\alpha_{eq} - \alpha^n)\frac{\epsilon}{F(\alpha^n)}.$$

One can show that $\beta > 0$ using the variations of F studied in the previous section. The solution to this new ODE is then given by:

$$\alpha_\beta^{n+1} = \alpha_{eq} + (\alpha^n - \alpha_{eq}) \exp(-\Delta t/\beta). \tag{5.12}$$

We can see that the value of α_β^{n+1} tends to the equilibrium value in long times $\Delta t \to \infty$, whatever the value of β.

5.1.3.4 Using the approximate solution

Another approach consists in using the approximate solution presented in paragraph 5.1.2.3. One has now multiple choices for α^*. Let us present two of them:

- if we know that the time step of integration is negligible compared to the characteristic relaxation time ($\Delta t \ll \tau_{\text{relax}}$), one expects α to stay close to α^n during the integration and one can choose $\alpha^* = \alpha^n$. From (5.9), the approximate solution then writes:

$$\alpha_A^{n+1} = \alpha^n + \frac{F(\alpha^n)}{F'(\alpha^n)} \left(\exp\left(\frac{F'(\alpha^n)\Delta t}{\epsilon} \right) - 1 \right) \tag{5.13}$$

  One can note that when $\Delta t \to \infty$, this solution tends to $\alpha^n - \frac{F(\alpha^n)}{F'(\alpha^n)}$ which is a priori different from α_{eq}, and could even be out of $]0, 1[$.

- in situations where α is initially close to α_{eq}, one would like to apply $\alpha^* = \alpha_{eq}$. The approximate solution would then be given by:

$$\alpha_B^{n+1} = \alpha_{eq} + (\alpha^n - \alpha_{eq}) \exp\left(\frac{F'(\alpha_{eq})\Delta t}{\epsilon} \right). \tag{5.14}$$

  For this solution, α tends to the equilibrium value in long times, but is not very accurate for short time-integration when α^n and α_{eq} differ significantly.

The two approximate solutions α_A and α_B are A-stable, but α_A is not L-stable. We can moreover note that the initial slopes of α_A and α_β solutions are equal: α_A solution seems then less interesting than α_β.

5.1.3.5 Implicit integrations

Let us finish this list of approximate solutions by presenting a first and a second-order Euler-type implicit integration methods for the ODE (5.3). These methods are known to be A-stable and L-stable. The first-order method is even *stiffly accurate* (Prothero and Robinson (1974)). Let us give some further details in the case of the resolution of (5.3).

The first order implicit Euler method applied to the ODE (5.3) comes down to solve:

$$\alpha^{n+1} = \alpha^n + \Delta t \frac{F(\alpha^{n+1})}{\epsilon}. \tag{5.15}$$

As there is no obvious solution to equation (5.15) (it necessitates to inverse a third order polynomial), one has to approximate the value of α^{n+1} using some root-finding algorithm. Let us prove that root-finding algorithms are applicable, by considering the series defined implicitly by (5.15), and by introducing the function $F_{\Delta t/\epsilon}$:

$$G_{\Delta t/\epsilon} : \begin{pmatrix}]0,1[& \to & \mathbb{R} \\ \alpha & \mapsto & \alpha - \Delta t \dfrac{F(\alpha)}{\epsilon} \end{pmatrix}$$

The series reads:

$$G_{\Delta t/\epsilon}\left(\alpha^{n+1}\right) = \alpha^n.$$

We have the following properties:

Proposition 5.

1. Function $G_{\Delta t/\epsilon}$ is strictly increasing over the domain $]0,1[$.
2. Equation (5.15) has a unique solution in $]0,1[$.
3. If $\alpha^n = \alpha_{eq}$, then $\alpha^{n+1} = \alpha_{eq}$.

From properties 1. and 2., one can deduce that root-finding algorithms are applicable in this situation. The third property means that if the initial value of α is at equilibrium, then the source term preserves the equilibrium.

Proof of Proposition 5. We have seen previously that $F'(\alpha) < 0$ whatever the value of $\alpha \in]0,1[$. Hence function $G_{\Delta t/\epsilon}$ is increasing over $]0,1[$. From the variations of F, we have also the following asymptotic limits:

$$G_{\Delta t/\epsilon} \xrightarrow[0^+]{} -\infty$$
$$G_{\Delta t/\epsilon} \xrightarrow[1^-]{} +\infty.$$

Consequently, the equation $G_{\Delta t/\epsilon}(\alpha) = \alpha^n$ has a solution, which is unique in $]0,1[$. Finally, since $G_{\Delta t/\epsilon}(\alpha_{eq}) = \alpha_{eq}$, because $F(\alpha_{eq}) = 0$, this concludes for the third property. □

Let us also consider a second order method of type BDF2 (*Backward Differentiation Formula*, Curtiss and Hirschfelder (1952)), which reads:

$$\begin{cases} \alpha^{n+1/2} &= \alpha^n &+ \frac{1}{2}\Delta t \frac{F\left(\alpha^{n+1/2}\right)}{\epsilon} \\ \alpha^{n+1} &= \frac{4}{3}\alpha^{n+1/2} - \frac{1}{3}\alpha^n &+ \frac{1}{3}\Delta t \frac{F(\alpha^{n+1})}{\epsilon} \end{cases} \tag{5.16}$$

By using the same reasoning as before, it is possible to prove that this method has the same good properties as the first order implicit method, and can be solved by using a root-finding algorithm.

5.1.4 Comparisons of the numerical solutions

Here, we propose to compare the solution obtained with the different methods presented in the previous section. We recall that $\Delta t = t - t^n$ is the time step of integration, and we compare the solution after one iteration of the following methods:

- α_{Exact} is the exact solution of the ODE (5.3),
- α_β is the solution of an exponential relaxation, and is given by (5.12),
- α_A from (5.13) is the solution of the approximate ODE, linearized in the vicinity of the initial condition,
- α_B from (5.14) is the solution of the approximate ODE, linearized in the vicinity of the equilibrium value,
- α_{BDF1} is the first-order implicit solution satisfying (5.15),
- α_{BDF2} is the second-order implicit solution satisfying (5.16).

Finally, we also present a solution made of a hybrid mixing between the approximate solutions, so as to get a good approximation for both small and large time step of integration Δt. It reads:

$$\alpha_{Strat}(t) = \begin{cases} \alpha_\beta(t) & \text{if } t < t_1, \\ \alpha_B + (\alpha_\beta(t_1) - \alpha_{eq}) \exp\left(\frac{F'(\alpha_{eq})(t-t_1)}{\epsilon}\right) & \text{if } t \geqslant t_1, \end{cases} \tag{5.17}$$

and t_1 is chosen such that $\alpha_\beta(t_1) = \gamma\alpha^n + (1-\gamma)\alpha_{eq}$ with $\gamma \in [0,1]$. The choice of γ could be made by estimating an optimal value. However, in a first and simple approach, we fix it to 0.5. Improving this strategy may be done in future works. Let us also note that $\alpha_{Strat} \to \alpha_{eq}$ when $t \to \infty$.

Remark 23. *This method is interesting only in the case where one time iteration is performed to integrate the ODE. Otherwise, high-order methods are preferable.*

First, we present some tests in a non-stiff configuration, meaning that the time step Δt is smaller than the physical relaxation time τ_{relax}. Next, we study stiffer cases, which are encountered where the initial value of α and/or of the equilibrium value α_{eq} are close to 0 or 1. These configurations are stiff due to the non-linearity of F in (5.3).

5.1.4.1 Smooth tests

Two smooth configurations are presented in Figure 5.2. The results are given as a function of the integration time step Δt, given in abscissa. In Figure 5.2a, the initial value of volume fraction is $\alpha^n = 0.9$ and the equilibrium is $\alpha_{eq} = 0.1$. For this configuration, we can observe that the closest solution to the exact solution, whatever Δt is the second order implicit solution α_{BDF2}.

Approximate solutions α_β and α_B are closer to α_{Exact} than α_A. The mixing strategy α_{Strat} enables to get close to the exact solution for large Δt. The first-order implicit solution *converges* slowly to the equilibrium (in the sense that for large Δt, it is less accurate than the other solutions), while α_A tends to a volume fraction value that is not the equilibrium.

The same observations can be done for the second smooth case, in Figure 5.2b, where $\alpha^n = 0.3$ and $\alpha_{eq} = 0.5$. For this case, the mixing solution α_{Strat} seems to give very good results.

5.1.4.2 Stiff configurations

The stiff configurations are defined as configurations where either α^n or α_{eq} are close to 0 or 1. In these cases, the relaxation rate can be very strong, leading also to very large coefficients in the approximate solutions α_β, α_A, α_B. Results are presented in Figure 5.3.

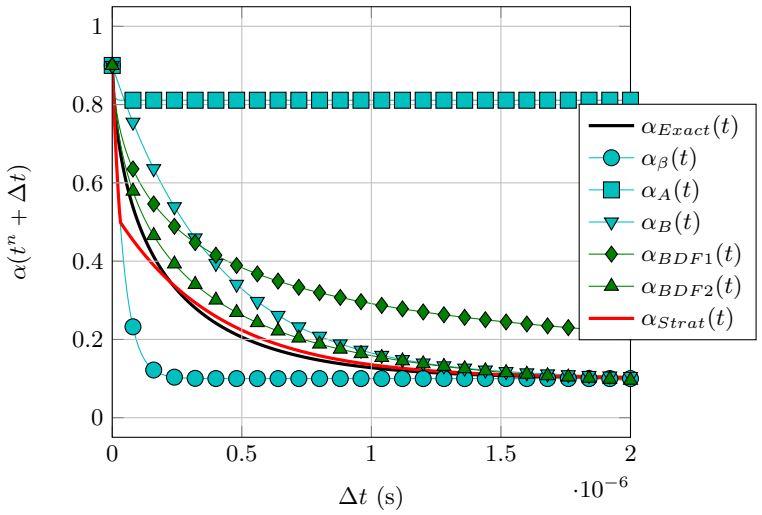
In the first configuration (Figure 5.3a), the equilibrium value α_{eq} is very small: $\alpha_{eq} = 1.0 \cdot 10^{-6}$. In this case, the approximate solution which is the closest to the exact solution is α_β. α_B has a too large decreasing rate, while α_A does not tend to the equilibrium value. The mixing strategy α_{Strat} does not seem to be of a good accuracy for this configuration, except for short Δt. The implicit solutions are also *far* from the exact solution: the first order α_{BDF1} has a slow decreasing rate towards equilibrium, while the second order α_{BDF2} underpasses the equilibrium value, and gets closer to 0. We recall that we compare the different solutions for one integration time step only. The behaviors of α_{BDF1} and α_{BDF2} does not mean that the methods are not converging, but are not as accurate as the other approximate solutions.

With the second configuration (Figure 5.3b), where the initial value is close to 1 ($\alpha^n = 1.0 - 1.0 \cdot 10^{-6}$), results seem better for the mixing strategy α_{Strat}, and now α_B is closer to the exact solution than α_β and α_A. Implicit strategies are also closer to the exact solution than for the configuration of Figure 5.3a. Here, the stiffness of the solution is due to the value of the initial solution α^n: this is why the approximate methods which are based on a Taylor expansion near α^n fail for this test.

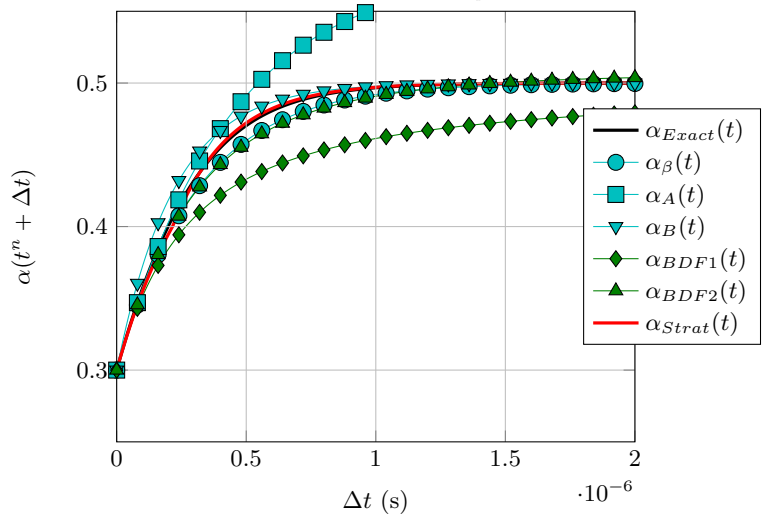
5.1.4.3 Conclusion

In conclusion to the previous observations, two options may be considered for the integration of the source term in the 4-equation system:

- the first one is to use the exact solution, defined implicitly. This is certainly the most reliable choice, and moreover it is appropriate for the use of root-finding algorithms. We could then envision using such computationally efficient methods.
- among the approximate solutions, evaluated on one time step Δt, the mixing solution α_{Strat} seems to be a promising strategy, provided that the coefficient γ is estimated more precisely than the fixed value 0.5. In non-stiff cases, it gives good results for a low computational cost. In stiff conditions, it is still stable.

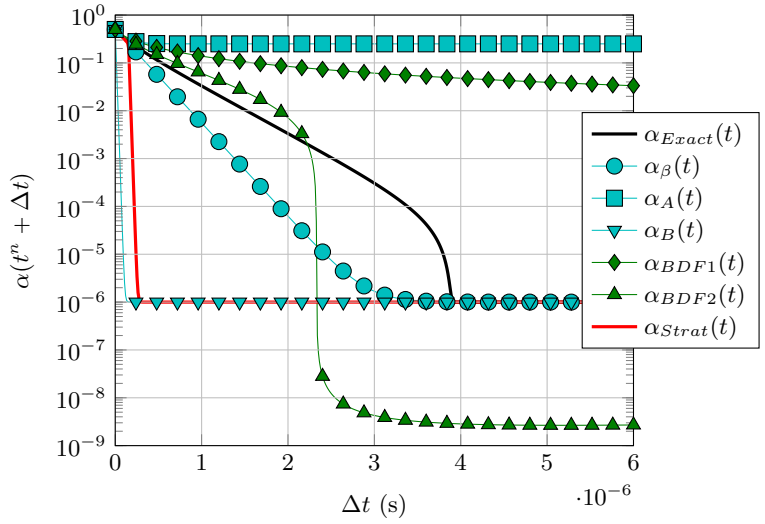


(a) Relaxation of α from $\alpha^n = 0.9$ to $\alpha_{eq} = 0.1$. $\epsilon = 1.0 \cdot 10^3$.

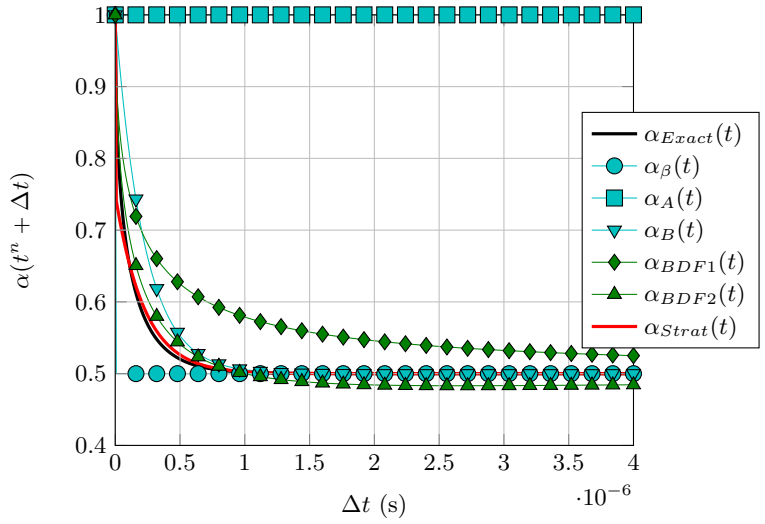


(b) Relaxation of α from $\alpha^n = 0.3$ to $\alpha_{eq} = 0.5$. $\epsilon = 1.0 \cdot 10^3$.

Figure 5.2: *Solutions for the integration of* (5.3) *given as a function of a unique integration time step* Δt. *Results for smooth cases: the four approximate solutions in cyan lines, the implicit integrations in green lines, the mixing strategy in red, the exact solution in black.*



(a) Relaxation of α from $\alpha^n = 0.5$ to $\alpha_{eq} = 1.0 \cdot 10^{-6}$, $\epsilon = 1.0 \cdot 10^{3}$.



(b) Relaxation of α from $\alpha^n = 1.0 - 1.0 \cdot 10^{-6}$ to $\alpha_{eq} = 0.5$, $\epsilon = 1.0 \cdot 10^{3}$.

Figure 5.3: *Solutions for the integration of* (5.3) *given as a function of a unique integration time step* Δt*. Results for stiff cases: the four approximate solutions in cyan lines, the implicit integrations in green lines, the mixing strategy in red, the exact solution in black.*

In future work, it would be interesting to consider high-order ODE solvers (Radau5, VODE...) and make a comparison in terms of computational costs with the computation of the exact solution. In particular, this computational cost may have a strong influence in a real-case simulation, where the source terms must be integrated in numerous cells.

5.2 Integration of the 5-equation oscillating source term

The source terms of the 5-equation system have effects on the equations on α and w. We recall here that in the second stage of the operator splitting process, the integration of the source terms lies in solving the following system of ODEs:

$$\begin{cases} \mathrm{d}_t\alpha &= \dfrac{\overline{\rho}\overline{Y}w}{\sqrt{\nu}} \\ \mathrm{d}_t w &= \dfrac{F(\alpha)}{\overline{\rho}\overline{Y}\sqrt{\nu}} - \dfrac{\epsilon}{\nu}w \end{cases} \tag{5.18}$$

where $F(\alpha)$ has been introduced in the previous section. Besides the fact that two variables are now concerned with the source terms, their nature is different from the relaxation source term of the 4-equation system. Indeed, while the relaxation source term was behaving similarly to an exponential decrease towards equilibrium, for the 5-equation system, this decrease is closer to a damped harmonic oscillator, as system (5.18) also reads:

$$\mathrm{d}_t^2\alpha + \frac{\epsilon}{\nu}\mathrm{d}_t\alpha - \frac{F(\alpha)}{\nu} = 0, \tag{5.19}$$

obtained by deriving the equation on α with respect to time in (5.18). In this context, $F(\alpha)$ can be physically interpreted as a non-linear stiffness.

Let us also introduce the vector $\mathbf{U} = (\alpha, w)^T$, and rewrite (5.18) as:

$$\mathrm{d}_t\mathbf{U} = \mathbf{G}(\mathbf{U}). \tag{5.20}$$

5.2.1 Study of the differential equation

Exact integration:

In Chapter 2, we have seen that the equations for α and for w are equivalent to the Rayleigh-Plesset's equation, that models the pulsations of bubbles. If the term $F(\alpha)$ in (5.19) were linear, this ODE would correspond to a damped harmonic oscillator. However, the non-linearity of this term has two non-negligible effects:

- it is a very strong return term that forces the volume fraction to stay within the interval $]0,1[$ whatever the initial conditions and the values of ϵ and ν (that can lead to very stiff configurations),
- near the limits $\alpha = 0$ and $\alpha = 1$, it may take arbitrarily large values, which lead to numerical difficulties in these zones (see for instance the test case of Appendix B where the stiffness of the ODE is extremely high).

An analytical solution to equation (5.19) is out of reach here. We can only show that in long times, when $\epsilon > 0$, $\alpha(t) \to \alpha_{eq}$, and $w \to 0$. For these reasons, we are interested in approximate solutions in the sequel.

Taylor expansion and approximate solution to the ODE:

As for the relaxation source term of the previous section, let us develop $F(\alpha)$ in a Taylor series near a chosen value α^*. The formula is given by (5.8). At first-order, the oscillating ODE (5.19) becomes:

$$\mathrm{d}_t^2\alpha + \frac{2}{\tau}\mathrm{d}_t\alpha + {\omega_0^*}^2\alpha + \frac{\alpha^* F'(\alpha^*) - F(\alpha^*)}{\nu} = 0, \tag{5.21}$$

with $\tau = 2\nu/\epsilon$, $\omega_0^{*2} = -F'(\alpha^*)/\nu$. The solution of this second order ODE is given by the classic solution of the damped harmonic oscillator. The integration from $t = 0$ to $t = t_f$ reads:

$$\alpha(t_f) = \alpha^* - \frac{F(\alpha^*)}{F'(\alpha)^*} + \left| \begin{array}{ll} \gamma^* \exp(-t_f/\tau) \sin(\theta^* + \omega^* t_f) & \text{if } \omega_0^* > 1/\tau, \\ \left(\left(\dfrac{L^*}{\tau} + M \right) t_f + L^* \right) \exp(-t_f/\tau) & \text{if } \omega_0^* = 1/\tau, \\ A^* \exp\left(-\dfrac{(1+\sqrt{1-\omega_0^{*2}\tau^2})t_f}{\tau} \right) + B^* \exp\left(-\dfrac{(1-\sqrt{1-\omega_0^{*2}\tau^2})t_f}{\tau} \right) & \text{if } \omega_0^* < 1/\tau, \end{array} \right. \tag{5.22}$$

with:

$$\gamma^* = \left(L^{*2} + \left(\frac{M + L^*/\tau}{\omega^*} \right)^2 \right)^{1/2}, \quad \theta^* = \arctan\left(\frac{\omega^* L^*}{M + L^*/\tau} \right), \quad \omega^* = \frac{\left(\epsilon^2 + 4\nu F'(\alpha^*)\right)^{1/2}}{2\nu},$$

$$L^* = \alpha(0) - \alpha^* + \frac{F(\alpha^*)}{F'(\alpha)^*}, \quad M = \frac{\overline{\rho}\overline{Y} w(0)}{\sqrt{\nu}},$$

$$A^* = \left[M + \frac{1 - \sqrt{1-\omega_0^{*2}\tau^2}}{\tau} L^* \right] \frac{\tau}{-2\sqrt{1-\omega_0^{*2}\tau^2}}, \quad B^* = \left[M + \frac{1 + \sqrt{1-\omega_0^{*2}\tau^2}}{\tau} L^* \right] \frac{\tau}{2\sqrt{1-\omega_0^{*2}\tau^2}}.$$

Let us note that the first part of the solution in (5.22) can also been written as:

$$\alpha(t_f) = \alpha^* - \frac{F(\alpha^*)}{F'(\alpha)^*} + \exp(-t_f/\tau) \left(L^* \cos(\omega^* t_f) + \frac{M + L^*/\tau}{\omega^*} \sin(\omega^* t_f) \right),$$

where the initial condition is obviously recovered when $t_f = 0$. We can also see that $\alpha \underset{t \to +\infty}{\longrightarrow} \alpha^* - \dfrac{F(\alpha^*)}{F'(\alpha)^*}$, with oscillations of pseudo-period ω^* and a decreasing rate of $\frac{1}{\tau}$. Both ω^* and τ depend on parameters ϵ and ν:

- when $\epsilon \ll \nu$, the characteristic damping time τ increases, while ω^* is closer to the natural frequency ω_0^*. The behavior of the solution is thus close to an undamped harmonic oscillator;
- when $\nu \ll \epsilon$, the characteristic damping time τ decreases and the pulsation ω^* increases. We get than a solution which is rapidly damped, but whose frequency of oscillations can be large.

The second case appears to be stiff, and can cause integration difficulties.

5.2.2 Numerical source integration strategies

As in the previous section, we consider various approximations of the solution of (5.20), which will be used in a one time step integration strategy of low-order accuracy:

- solutions based on the exact resolution of a damped harmonic oscillator-like ODE (5.21),
- implicit strategies consisting in solving the system (backward Euler method):

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \mathbf{G}(\mathbf{U}^{n+1}), \tag{5.23}$$

or at higher orders, like the backward differential formulae Curtiss and Hirschfelder (1952), in solving:

$$\sum_{k=0}^{s} a_k \mathbf{U}^{n+k} = \beta \frac{\Delta t}{s} f(\mathbf{U}_{n+s}). \tag{5.24}$$

Remark 24. *Contrary to the previous section, the exact solution to (5.19) is out of reach here. For the comparisons of the methods, we will take the solution of a high-order method, namely VODE solver (Brown, Byrne, and Hindmarsh (1989b)) as a reference solution. A consequence is that we can not study very stiff cases, since the VODE solver fails.*

Let us now present more in detail the different approaches for integrating (5.18). We again integrate between times $t = t^n$ and $t = t^{n+1} = t^n + \Delta t$ and note $\alpha(t^n) = \alpha^n$, $\alpha(t^{n+1}) = \alpha^{n+1}$.

5.2.2.1 Solutions based on the approximate equation

We consider here the approximate solution (5.22) for two different choices of α^*.

Solution for $\alpha^* = \alpha^n$:

Assuming that the integration time step is small compared to the damping characteristic time, $\Delta t \ll \tau$, one can use $\alpha^* = \alpha^n$. The approximate solution is then given by (5.22), where in particular

$$L^* = \frac{F(\alpha^n)}{F'(\alpha^n)}$$

The main drawback of this option is that the so-defined solution $\alpha_A(t)$ does not tend to α_{eq} for large integration time steps Δt.

Solution for $\alpha^* = \alpha_{eq}$:

When the initial volume fraction α^n is close to the equilibrium value α_{eq}, and the variable w is also close to 0 (little pulsation effects), one can choose $\alpha^* = \alpha_{eq}$. The approximate solution given by (5.22) has then the following specific parameters:

$$F(\alpha_{eq}) = 0\,, \quad L^* = \alpha(0) - \alpha_{eq}.$$

This so defined solution α_B has a good long time behavior, as it tends to the equilibrium value, but may be less accurate than α_A at very short times when α remains close to $\alpha(t^n)$.

Defaults of these approximate solutions:

Although these approximate solutions have the advantage of using approximate analytical formulations, we will see below with the different tests that they hardly handle stiff cases. These cases occur when α_{eq} or α^n is close to 0 and 1, and the initial decreasing or increasing rate of the solution is large. This may be due to a large difference between α^n and α_{eq}, which implies large values for L^* and γ^*. In the same way, large values of w^n make the parameter M be large too. The approximate solutions are then going out of the interval $]0, 1[$, until the damping effect brings them back closer to the equilibrium value.

5.2.2.2 Implicit integrations

As for the 4-equation source term, we present here the implementation of Backward Differentiation Formulae for implicit integration. They are known to be A-stable and are easy to use. Moreover, they are also more stable for integration over small time steps. The drawbacks of these methods may be their computational cost and their lower accuracy, as we will see when comparing with the approximate solutions of the previous paragraph.

For the first oder implicit method, let us consider the following system:

$$\begin{cases} \alpha^{n+1} &= \alpha^n &+ \Delta t \dfrac{\rho Y w^{n+1}}{\sqrt{\nu}} \\ w^{n+1} &= w^n &+ \Delta t \dfrac{F(\alpha^{n+1})}{\rho Y \sqrt{\nu}} \quad - \quad \Delta t \dfrac{\epsilon}{\nu} w^{n+1} \end{cases}$$

which, by rearranging the different terms, is equivalent to:

$$\begin{cases} \alpha^{n+1} &= \alpha^n &+ \Delta t \dfrac{\rho Y \sqrt{\nu}}{\nu + \epsilon \Delta t} \left(w^n + \Delta t \dfrac{F(\alpha^{n+1})}{\rho Y \sqrt{\nu}} \right) \\ w^{n+1} &= & \left(w^n + \Delta t \dfrac{F(\alpha^{n+1})}{\rho Y \sqrt{\nu}} \right) \left(1 + \Delta t \dfrac{\epsilon}{\nu} \right)^{-1} \end{cases} \tag{5.25}$$

Then, we can solve the first equation in (5.25) to get α^{n+1}, and then the second one for w^{n+1}.

Let us note that the first equation can be written in the form:

$$\alpha^{n+1} = \alpha^n + C^n + \Delta t^2 \frac{F(\alpha^{n+1})}{\nu + \epsilon \Delta t},$$

which is very close to the implicit formula for the 4-equation source term. The same demonstrations could be used to prove the first two properties of Proposition 5. However, if $w^n \neq 0$, the third point of Proposition 5 is not verified, as pulsations described by w bring the system out of equilibrium.

The principle of the second order BDF is the same as for the 4-equation system. It consists in solving the following system:

$$\begin{cases} \alpha^{n+1/2} &= \alpha^n &+ \dfrac{1}{2} \Delta t \dfrac{\rho Y w^{n+1/2}}{\sqrt{\nu}} & & \\ w^{n+1/2} &= w^n &+ \dfrac{1}{2} \Delta t \dfrac{F(\alpha^{n+1/2})}{\rho Y \sqrt{\nu}} & - \dfrac{1}{2} \Delta t \dfrac{\epsilon}{\nu} w^{n+1/2} \\ \alpha^{n+1} &= \frac{4}{3}\alpha^{n+1/2} - \frac{1}{3}\alpha^n &+ \frac{1}{3} \Delta t \Delta t \frac{\rho Y w^{n+1}}{\sqrt{\nu}} & \\ w^{n+1} &= \frac{4}{3} w^{n+1/2} - \frac{1}{3} w^n &+ \dfrac{1}{3} \Delta t \dfrac{F(\alpha^{n+1})}{\rho Y \sqrt{\nu}} & - \dfrac{1}{2} \Delta t \dfrac{\epsilon}{\nu} w^{n+1} \end{cases} \tag{5.26}$$

5.2.3 Comparisons of the numerical solutions

In this part, we compare the different approaches presented previously, for the integration of (5.18):

- α_A the approximate solution given by (5.22) with $\alpha^* = \alpha^n$,
- α_B the approximate solution given by (5.22) with $\alpha^* = \alpha_{eq}$,
- α_{BDF1} the solution obtained with the first order implicit integration satisfying (5.25),
- α_{BDF2} the solution obtained with the second order implicit integration satisfying (5.26).

A reference solution is provided by VODE solver (from the Scipy.integrate package), integrated through time with 1000 time steps. The solutions we compare, however, are integrated on only one time step Δt, and the results are given as a function of Δt.

In a first time, we study smooth cases, namely cases with initial conditions close enough to equilibrium (defined by $\alpha^n = \alpha_{eq}$ and $w^n = 0$) so that the approximate solutions do not leave $]0, 1[$.

For very stiff cases, VODE solver may fail to converge (as does Radau5 in Appendix B). The tests we are presenting below are not too stiff for these methods to work, but stiff enough to show the limits of our approximate solutions. The question of solving extremely stiff configurations is still open, and we propose at the end of this section a naive numerical strategy that ensures $\alpha \in]0, 1[$ for a reasonable cost.

5.2.3.1 Smooth cases

We present three smooth configurations, corresponding to the three regimes of the damped harmonic oscillator:

- in the first regime (see Figure 5.4a), one observes damped oscillations. The approximate solution α_B is very close to the reference solution in long times, and only slightly underestimates the damping

in short times. The approximate solution α_A is not as accurate as α_B: the period of its oscillations is different and it does not tend to the equilibrium state. Both implicit methods do not develop the oscillating behavior, but tend to the equilibrium in long times.

- in Figure 5.4b, the solution stands close to the *critical* regime defined by $\omega_0^* = \frac{1}{\tau}$. The approximate solution α_B is again the closest solution to the reference one for this test. The approximate solution α_A does not converge towards the equilibrium state. The second-order implicit method is closer than in the first test, but is not accurate in long integration times.
- these observations are the same for the third regime, namely the *overdamped* regime presented in Figure 5.4c.

5.2.3.2 Stiff cases

In the stiff cases, the initial conditions (α^n and w^n) are such that the real solution goes very close to the borders of the admissible values for α which are 0 and 1. While the reference solution is maintained inside $]0,1[$ due to the non-linear effects, the approximate solution is not in most cases. From now on, we do not consider α_A, since it gives less accurate results than α_B in most cases.

We can observe these defects of the approximate solution α_B in Figure 5.5a, where the volume fraction equilibrium value is close to 1, and the initial condition is such that $1 - \alpha^n = 4(1 - \alpha_{eq})$. Then, the approximate solution α_B is lead beyond 1. However, we can see that the decreasing rate and the period of oscillations seem close to the reference solution. As for the smooth case, the implicit solutions are not oscillating, but stay below 1.

In Figure 5.5b, the volume fraction stays in the neighborhood of the value 0. Although at initial state, $\alpha^n = \alpha_{eq}$, a large value of w^n generates strong oscillations. In the approximate solution α_B, values of α become negative, but again, periods of oscillations and damping are close to the reference solution. The implicit solutions still give poor results, although the volume fraction stays positive. We also show in Figure 5.5c the evolution of the variable w, which is well-approximated by α_B.

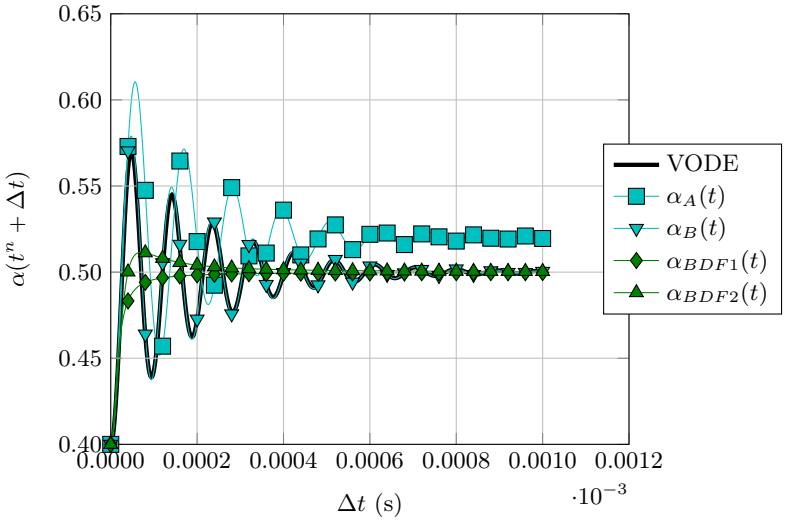
5.2.3.3 Conclusion and towards a final strategy

Let us first recall that the tests presented above are chosen so that VODE solver can work. There are indeed many very stiff configurations where this solver fails. For such situations, we are looking for a robust method, i.e. that is able to keep the value of α in its set of admissible values $]0,1[$. With the objective of developing such methods, we have studied in this section two main approaches for the integration of the ODE (5.18) (or equivalently (5.19)):

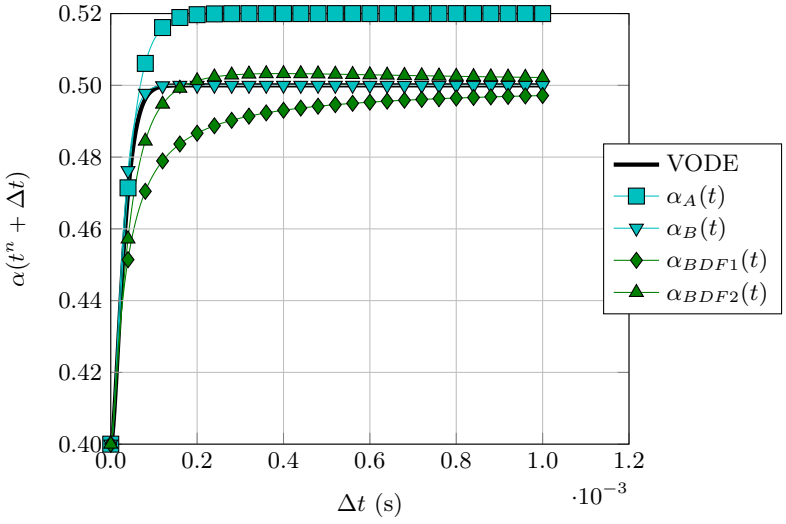
- integrations based on the exact solution of an approximation of (5.19), issued from a Taylor expansion in the neighborhood of either the initial solution (this is the approximate solution α_A) or the equilibrium value (this is the approximate solution α_B);
- standard first and second order implicit Euler methods: α_{BDF1} and α_{BDF2} respectively.

These methods have been compared for an integration over one time step Δt (without sub-steps) for smooth and stiffer configurations. The approximate method α_B provides an integration close to the exact solution, but with values of α that can leave the interval $]0,1[$, especially for small integration times Δt where the equation (5.19) is stiff. The first and second-order implicit integrations guarantee $\alpha \in]0,1[$, but their accuracy over one iteration is lower than the approximate solutions. In practice, for the rest of the present study, we propose to use the approximate solution α_B with a clipping process when it gives unacceptable values for α: in these cases, α is set to ξ or $1 - \xi$, with $\xi > 0$ small enough, but we keep the value of w.

Before considering higher-order methods in future works, it seems necessary to better understand the origin of the stiffness in (5.19), and study the influence of the source term integration in practical applications. In particular, we use α_B in the study of the numerical dispersion relations in the next section. The same study could be done with different ODE solvers and the results could be compared.

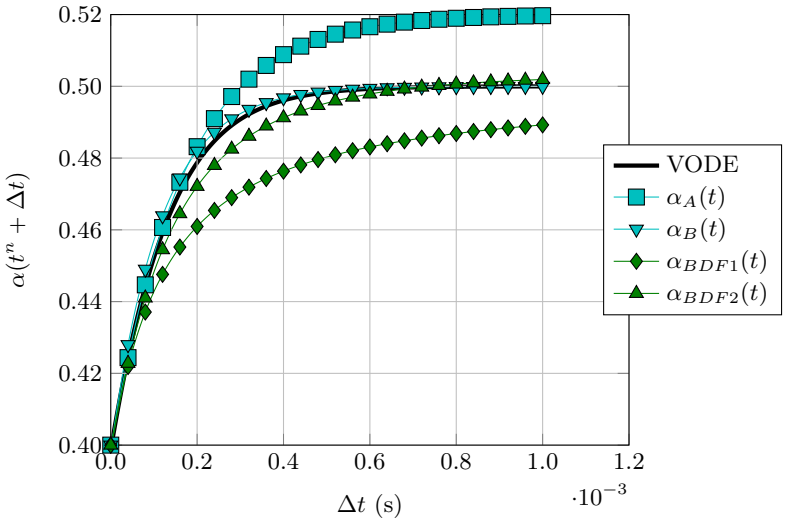


(a) Long time integration and oscillatory regime: $\nu = 1.0$, $\epsilon = 1.0 \cdot 10^4$.



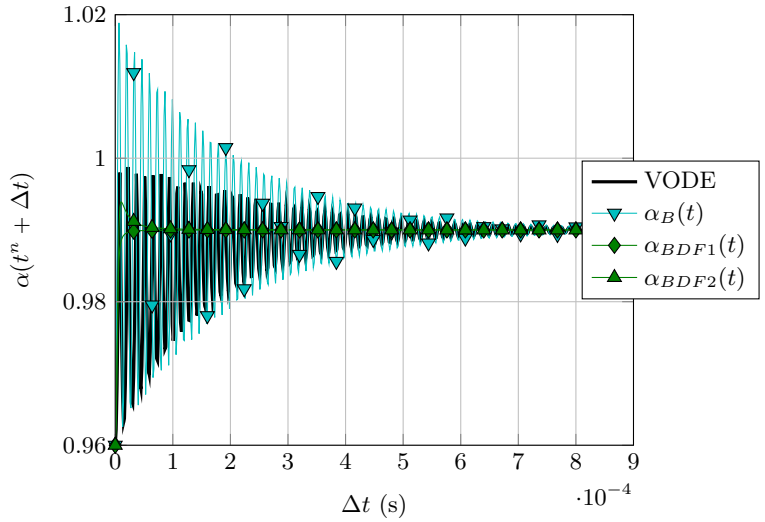
(b) Long time integration and critically damped regime: $\nu = 1.0$, $\epsilon = 1.3 \cdot 10^5$.

Figure 5.4: *Comparison of the approximate solutions α_A and α_B in cyan, the implicit solutions in green: first-order with diamonds, second-order with triangles. Integration over one time step Δt. The reference solution for the integration of (5.18) is given by a VODE solver in black. $\alpha_{eq} = 0.5$; $\alpha^n = 0.4$*



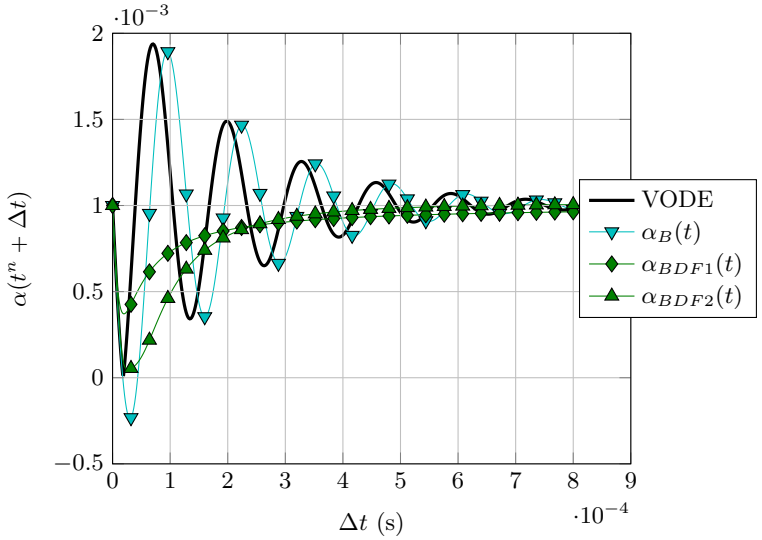
(c) Long time integration and overdamped regime: $\nu = 1.0$, $\epsilon = 5.3 \cdot 10^5$.

Figure 5.4: *(continued) Comparison of the approximate solutions α_A and α_B in cyan, the implicit solutions in green: first-order with diamonds, second-order with triangles. Integration over one time step Δt. The reference solution for the integration of (5.18) is given by a VODE solver in black. $\alpha_{eq} = 0.5$; $\alpha^n = 0.4$*

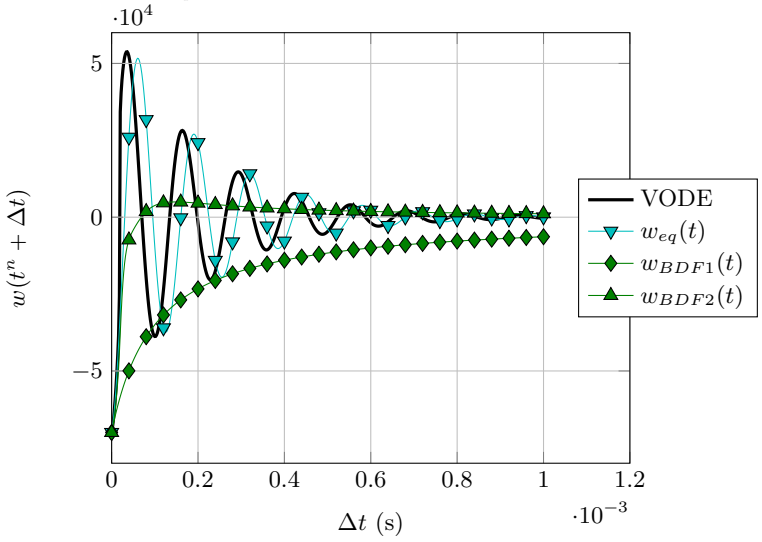


(a) Stiff initial condition which leads to $\alpha > 1$ for the approximate solutions: $\alpha^n = 1.0 - 4.0 \cdot 10^{-2}$, $\alpha_{eq} = 1.0 - 1.0 \cdot 10^{-2}$. $\nu = 1.0$, $\epsilon = 1.0 \cdot 10^4$.

Figure 5.5: *Comparison of the approximate solution α_B in cyan, the implicit solutions in green: first-order α_{BDF1} with diamonds, second-order α_{BDF2} with triangles. Integration over one time step Δt. The reference solution for the integration of (5.18) is given by a VODE solver in black.*



(b) Stiff initial condition which leads to $\alpha < 0$ for the approximate solutions: $\alpha^n = 1.0 \cdot 10^{-3}$, $\alpha_{eq} = 1.0 \cdot 10^{-3}$, $w(t^n) = -7.0 \cdot 10^4$. $\nu = 1.0$, $\epsilon = 1.0 \cdot 10^4$.



(c) Stiff initial condition which leads to $\alpha < 0$ for the approximate solutions.

Figure 5.5: *(continued) Comparison of the approximate solution α_B in cyan, the implicit solutions in green: first-order α_{BDF1} with diamonds, second-order α_{BDF2} with triangles. Integration over one time step Δt. The reference solution for the integration of (5.18) is given by a VODE solver in black.*

5.3 Numerical tests of the whole discretization strategy

In Chapter 4 and in the first part of this chapter, we have proposed methods to numerically solve the two-phase systems (4.1). Let us recall here the global strategy:

- the convective and source term parts of the equations are solved in successive steps, according to an operator splitting method (presented in the introduction of Part II);
- the convective part, which is hyperbolic, is solved numerically with an approximate Riemann solver Relax-B and an appropriate discretization of the transport equations, as presented in Chapter 4, section 4.2;
- the source terms are integrated by using approximate solutions of the ODEs, which have just been presented in this chapter, section 5.1 and 5.2.

This strategy is applied to all systems that were derived in Chapter 1, section 1.4. We have already tested the accuracy and the robustness of the integration of the convective part, as well as the integration of the ODEs (for which, strategies to ensure $\alpha \in]0,1[$ have been proposed). A first question is: "**What are the accuracy and the robustness of the whole strategy?**".

Another aspect tested in this section is the ability of the numerical schemes to preserve the asymptotic limits of the solutions. Indeed, from a modeling point of view, we know that the 3-equation system is the equilibrium limit of the 4-equation system when the parameter ϵ tends to zero. When considering a pressure perturbation in the form of an acoustic wave, this relation between the two systems is also observed through the phase velocity and spatial attenuation of the wave: when ϵ decreases, the phase velocity shifts from the Frozen value (c_{Frozen}) to the Wood's one (c_{Wood}), while attenuation decreases. In the same way, for a fixed value of ϵ, when ν tends to zero, the dispersion relations of the 5-equation system uniformly tend to those of the 4-equation one. These asymptotic transitions have been studied analytically in Chapter 2, section 2.4. The second question addressed in this section is: "**Do the numerical schemes preserve the asymptotic behaviors of the models?**".

5.3.1 Issues in the resolution of low-frequency acoustic waves

In this section, we present an issue which is related to the resolution of the 4-equation system in the limit $\epsilon \to 0$: the relaxation numerical scheme that was presented in Chapter 4, section 4.2 is indeed very diffusive in this asymptotic limit. Thus the expected asymptotic solution of the 4-equation system, which should be close to the solution of the 3-equation, shows an attenuation of the acoustic waves that is too high. In order to illustrate this issue, we present first a series of tests, and try to determine the origin of the problem. Then, we perform an analysis of the truncation error for the relaxation scheme applied to the 4-equation system, and show that in low-frequency limits, the numerical dispersion relations differ from the analytical ones. We finally propose a first modified scheme, which improves the accuracy of the solution for the tests that are presented here, but is not always stable, as we will see in the next sections.

5.3.1.1 First observations from the test of the propagation of a pressure perturbation:

Let us consider a periodic 1D domain on which the following initial conditions apply:

$$\begin{cases} p(0,x) &= f(x) \\ \alpha(0,x) &= 1.0 \cdot 10^{-2} \\ u(0,x) &= 0 \end{cases} \tag{5.27}$$

and:

$$f(x) = \begin{cases} p_0 + \sin\left(\dfrac{2\pi(x-0.5)}{0.1}\right) & \text{if} \quad x \in [0.45, 0.55] \text{ m}, \\ p_0 & \text{else.} \end{cases}$$

with $p_0 = 1.0 \cdot 10^5$. We also assume the pressure equilibrium: $p_1(0,x) = p_2(0,x)$. The space step of the discretization is $\Delta x = 5.0 \cdot 10^{-3}$ m. The Relax-B numerical scheme is used to solve the convective part of each model, and α_B ODE solver is used for the integration of the source term in the 4-equation system.

For times $t > 0$, the pressure perturbation will give rise to two new waves, whose magnitudes are twice smaller than the initial one. These waves will propagate in the domain at velocities $-c$ and c, where c can be c_{Wood}, c_{Frozen} or c_{Puls}, according to the model, the values of ϵ and ν and of the signal wavelength. They will also be attenuated, according to the model and to the values of the parameters. For instance, in Figure 5.6a, we show the pressure field $p(t, x)$ at time $t_f = 2.0 \cdot 10^{-4}$ s for both the 3-equation model and the 4-equation model with a very large value of ϵ ($\epsilon = 1.0 \cdot 10^{15}$): we can observe that the wave from the 3-equation model propagates at $c = c_{Wood}$ and the wave from the 4-equation model propagates at $c = c_{Frozen}$. Both waves are little attenuated. This is in agreement with the analytical dispersion relations in this regime (of frequency and for the value of ϵ). Now, let us study the case with $\epsilon \ll 1$.

Solutions obtained with the original schemes:

In Figure 5.6b, one can see a comparisons of the 3-equation and 4-equation pressure fields for this test at a time $t = 5.0 \cdot 10^{-3}$ s and with $\epsilon = 1.0 \cdot 10^{-15}$. Here, we expect the 4-equation solution to be close to the 3-equation one, since $\epsilon \ll 1$: the waves should propagate at $c = c_{Wood}$ and be little attenuated. However, this is not the case: the wave from the 4-equation model is extremely attenuated and it is thus difficult to determine its propagation velocity. By comparing the results of the two simulation, we have noted that the time steps were really different. Thus, we can suspect a harmful numerical diffusion effect. Let us explore this track.

Solution with a different CFL constraint:

Still considering the same initial conditions, the same models and numerical schemes, let us just change the time steps in the computation of the 3-equation solution: we choose to consider the CFL constraint by using c_{Frozen} instead of c_{Wood}. This would also come down to take a CFL coefficient of about 0.07 for this special test. The rest of the scheme is kept unchanged, as well as the scheme for the 4-equation system. The solutions for the 3-equation system with modified CFL constraint and for the 4-equation system are presented in Figure 5.7a: we can note that the 3-equation solution is more attenuated than in Figure 5.6b, but is still significantly less attenuated than the 4-equation solution.

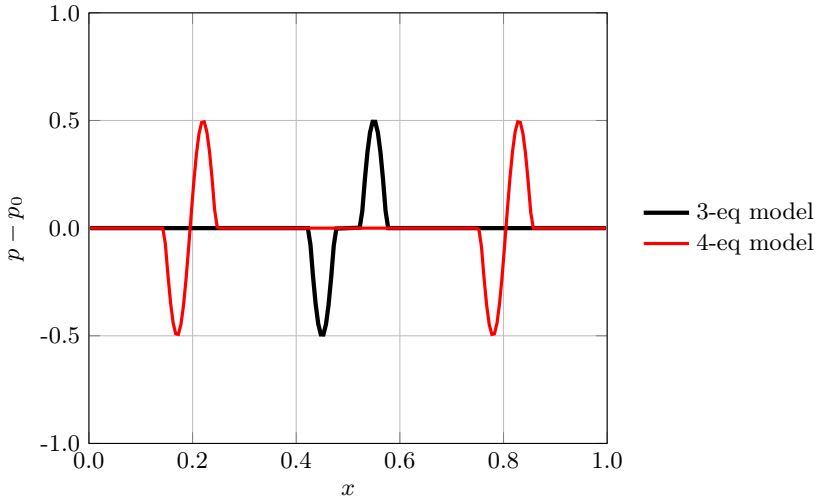
Solution with different characteristic velocities:

Let us also change the value of a inside the Riemann problem for the 3-equation: we recall that a was defined in Chapter 4, section 4.2 and is more or less equal to ρc, with c the characteristic velocity (c_{Wood}, c_{Frozen} or c_{Puls}). Then, for the 3-equation system, we choose to compute a according to c_{Frozen} instead of c_{Wood}. The time steps computation is consequently modified, and comes down to the same CFL constraint as in the previous paragraph. The rest of the scheme is kept unchanged, as well as the scheme for the 4-equation system. The solutions for the 3-equation system with modified a and for the 4-equation system are presented in Figure 5.7b. One can now see that the dissipation rates are of the same order for the 3-equation and 4-equation solutions: both solutions are too attenuated.

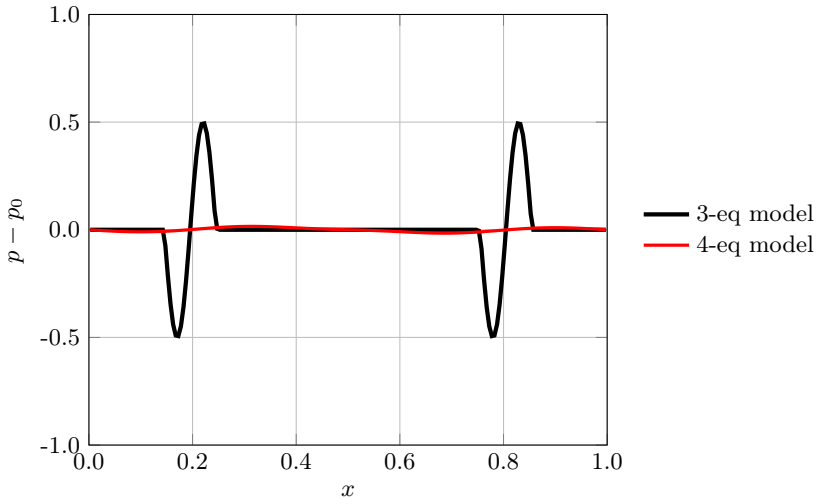
The last test consists in going back to the original scheme for the 3-equation system, and change the computation of a in the scheme for the 4-equation system. Now a is computed according to c_{Wood} instead of c_{Frozen}. One can note, from Figure 5.7c where the results are presented, that:

- the 4-equation solution is still stable, although the subcharacteristic condition is not satisfied for Relax-B associated to the 4-equation system,
- the dissipation rates of both models are finally of the same order of magnitude, and the wave is little attenuated.

To conclude briefly from these tests and observations, we can say that the dissipation in the 4-equation system solution seems to be related to the value of a, which is computed using c_{Frozen} in the original scheme Relax-B: the value of c_{Frozen} ($\approx$ 1500 m/s) is used whatever the wavelength of the acoustic perturbation and whatever the value of ϵ. However in configurations similar to the test presented here, the solution of the 4-equation system should be close to the solution of the 3-equation system, according to the analytical dispersion relations. There, the characteristic velocity is c_{Wood} ($\approx$ 107 m/s). In Appendix C, we propose an analysis of the truncation errors when the 4-equation system is solved with

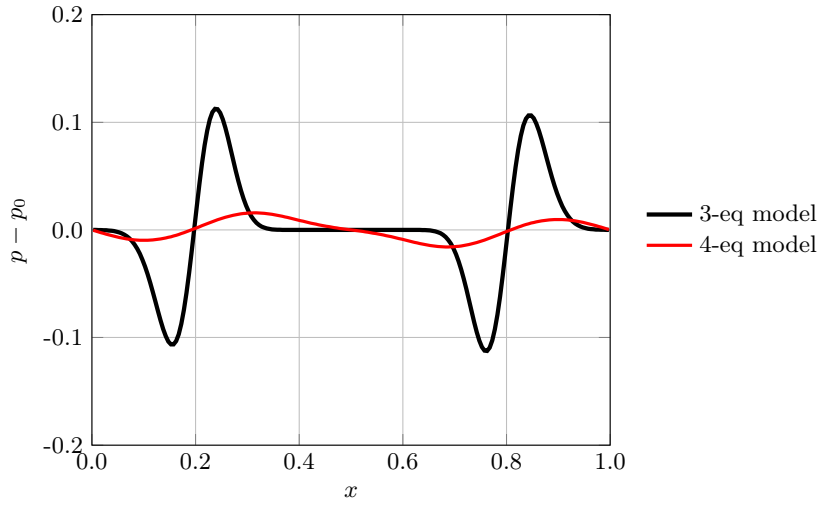


(a) Solution at time $t = 2.0 \cdot 10^{-4}$ s; $\epsilon = 1.0 \cdot 10^{15}$.

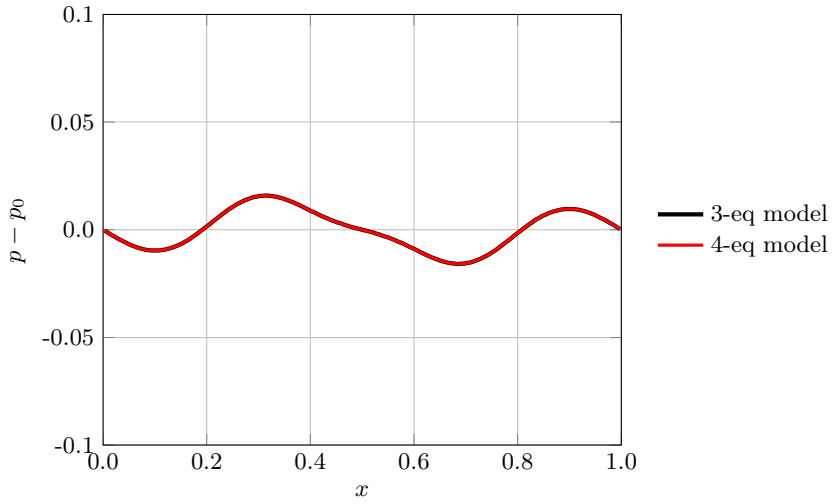


(b) Solution at time $t = 5.0 \cdot 10^{-3}$ s; $\epsilon = 1.0 \cdot 10^{-15}$.

Figure 5.6: *Propagation of an acoustic wave in a gas/liquid mixture, with* $\alpha = 1.0 \cdot 10^{-2}$.

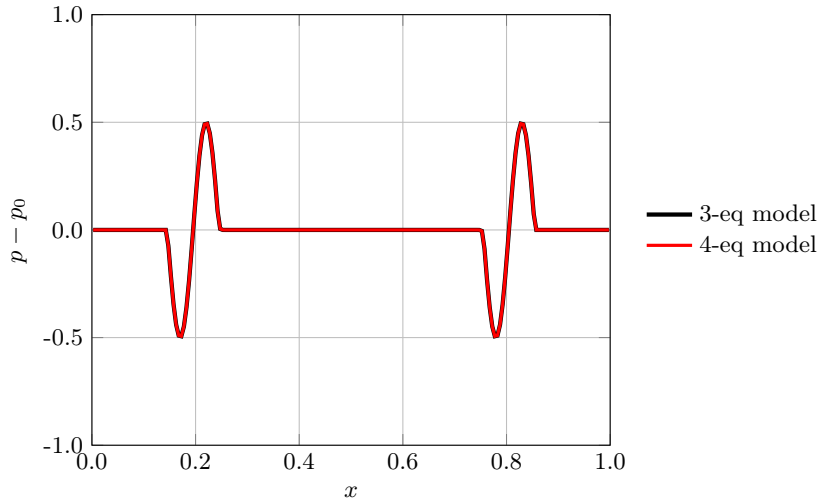


(a) Solution at time $t = 5.0 \cdot 10^{-3}$ s; $\epsilon = 1.0 \cdot 10^{-15}$. CFL condition for the 3-equation system is ≈ 0.07.

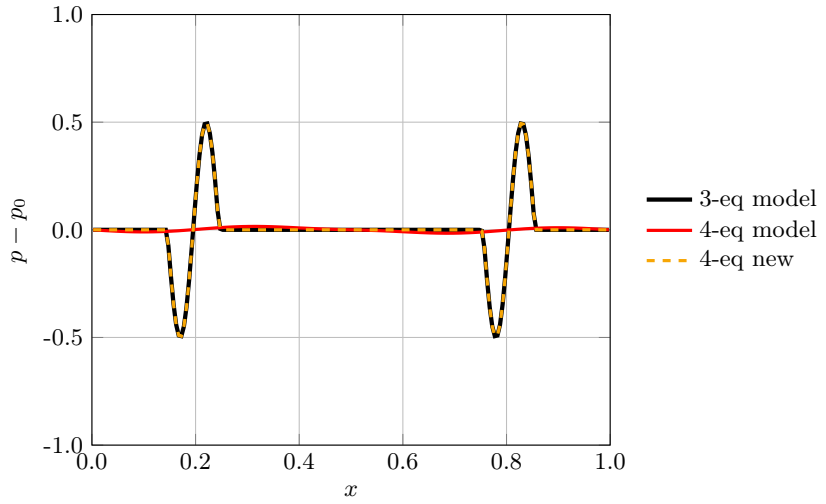


(b) Solution at time $t = 5.0 \cdot 10^{-3}$ s; $\epsilon = 1.0 \cdot 10^{-15}$. The value of a in the 3-equation Riemann's solver is chosen as $a = \rho c_{\text{Frozen}}$.

Figure 5.7: *Propagation of an acoustic wave in a gas/liquid mixture, with* $\alpha = 1.0 \cdot 10^{-2}$. $N = 200$.



(c) Solution at time $t = 5.0 \cdot 10^{-3}$ s; $\epsilon = 1.0 \cdot 10^{-15}$. The value of a in the 4-equation Riemann's solver is set to $a = \rho c_{\text{Wood}}$.



(d) Solution at time $t = 5.0 \cdot 10^{-3}$ s; $\epsilon = 1.0 \cdot 10^{-15}$. Yellow curve corresponds to the modified Bouchut's scheme, using $a = \rho c_{\text{Phase}}(2\Pi/\Delta t)$.

Figure 5.7: *(continued) Propagation of an acoustic wave in a gas/liquid mixture, with* $\alpha = 1.0 \cdot 10^{-2}$. $N = 200$.

a Lax-Friedrich's flux, in the limit of small values of ϵ. This study deserves improvement and should be done for Relax-B. However, it seems that the spatial attenuation of acoustic waves in the regime $\epsilon \ll 1$ is artificially increased by numerical diffusion. This has lead us to consider a modified scheme, which is presented below.

5.3.1.2 Proposition of correction for Relax-B

We propose to adapt the value of a, used in Relax-B scheme according to the analytical dispersion relations of the 4-equation system, which provide the sound speed c_{Phase} as a function of the wave pulsation ω.

Numerically, the pulsation ω could be related to the numerical time step $\omega = 2\pi/\Delta t$. Then we propose to modify Relax-B scheme of Chapter 4, section 4.2, such that $a = \rho c_{Phase}(2\pi/\Delta t)$. In the configuration of the test from the previous paragraphs, this was done with the choice of $a = \rho c_{Wood}$ (see results in Figure 5.7d) in the limit $\epsilon \ll 1$.

One can note that this corrected scheme has a default: when Δx decreases, due to the CFL constraint, Δt decreases too, while the physical pulsation of the pressure perturbation is kept unchanged. So, the value of ω is not really representative of the pressure waves, and for values of Δx small enough, one might recover the strong dissipation rate of the 4-equation solution.

Up to now, we have not mentioned the 5-equation model and its associated scheme. In practice (for the tests presented thereafter), we have not noticed any problem in the limit $\nu \to 0$: in this limit, both 4-equation and 5-equation solutions are the same. However, we recover the same problem as for the 4-equation system when $\epsilon \to 0$. Thus, we can propose a similar modification of Relax-B: we compute the value of a according to a characteristic frequency ω and the acoustic phase velocity $c_{Phase}(\omega)$ issued from the analytical dispersion relations. This correction is however far from being a satisfying solution: in the next section we will encounter configurations where the modified scheme is unstable.

5.3.2 Tests of acoustic waves propagation and numerical dispersion relations

Let us present here two concrete applications of the numerical schemes that have been presented so far:

- the first test consists in studying the propagation of an acoustic wave, when it passes from one medium to another, namely from a pure liquid phase to a gas/liquid mixture. Due to the difference of the acoustic impedances of these two media, which depend on the models, and on the values of parameters ϵ and ν, a part of the wave is reflected at the interface between these media, while another part is transmitted;
- from the test of a propagating acoustic wave in a mixture medium (in the same way as for the test of the previous paragraphs), we compute the propagation velocity and spatial attenuation of the wave for different values of the parameters ϵ and ν. Thus we draw the dispersion relations of the numerical schemes.

For all the tests, we use both Relax-B scheme and the modified version of Relax-B proposed in the previous section. The source terms are integrated using the approximate ODE solvers from the first part of this chapter.

5.3.2.1 Study of the acoustic impedance

Let us consider a 1D domain $[0, 1]$ with boundary conditions at infinity and on which the following initial conditions apply:

$$\begin{cases} p(0,x) &= f(x) \\ \alpha(0,x) &= g(x) \\ u(0,x) &= 0 \end{cases} \tag{5.28}$$

with:

$$f(x) = \begin{cases} p_0 + \sin\left(\dfrac{2\pi(x-0.5)}{0.1}\right) & \text{if} \quad x \in [0.45, 0.55] \text{ m}, \\ p_0 & \text{else.} \end{cases}$$

and:

$$g(x) = \begin{cases} 1.0 \cdot 10^{-10} & \text{if} \quad x < 0.7 \text{ m}, \\ 1.0 \cdot 10^{-4} & \text{else.} \end{cases}$$

with $p_0 = 1.0 \cdot 10^5$. We also assume the pressure equilibrium: $p_1(0,x) = p_2(0,x)$, and no pulsation $w(0,x) = 0$. The space step of the discretization is $\Delta x = 5.0 \cdot 10^{-3}$ m.

The left part of the domain ($x < 0.7$ m) can be considered as pure liquid, while the right part is a mixture made of a small volume fraction of gas. The initial condition, as well as the propagation of the pressure wave is illustrated in Figure 5.8 for the 3-equation model. We can see that, when the wave reaches the interface between the two media ($x = 0.7$ m), one part of the incident wave is reflected and propagates in the opposite direction, while another part is transmitted in the mixture medium, and propagates at c_{Wood} velocity.

The proportion γ_r of the incident wave magnitude that is reflected and the proportion γ_t that is transmitted are related to the acoustic impedances Z_L, Z_R of the left and right media respectively, by:

$$\gamma_r = \frac{Z_1 - Z_2}{Z_1 + Z_2},$$
$$\gamma_t = \frac{2Z_1}{Z_1 + Z_2}.$$

For fluid media, the acoustic impedance is given by $Z = \rho c_{Phase}$.

The comparison of the solutions for the three models and different values of ϵ and ν are presented in the different figures of Figure 5.9. For all these tests, one can note little improvement of the corrections proposed in the last section. This is due to the presence of the liquid phase, whose characteristic velocity is the most constraining for the time step computation. Moreover, the value of this velocity does not change with the models.

In Figure 5.9a and Figure 5.9d, one stands close to the equilibrium ($\epsilon \ll 1$ and $\nu \ll 1$) and all models behave as the 3-equation model.

In Figure 5.9b and Figure 5.9e, ϵ has an intermediate value, while $\nu \ll 1$. One can see that the 5-equation model behaves like the 4-equation model: the wave propagates faster in the mixture medium than c_{Wood} velocity.

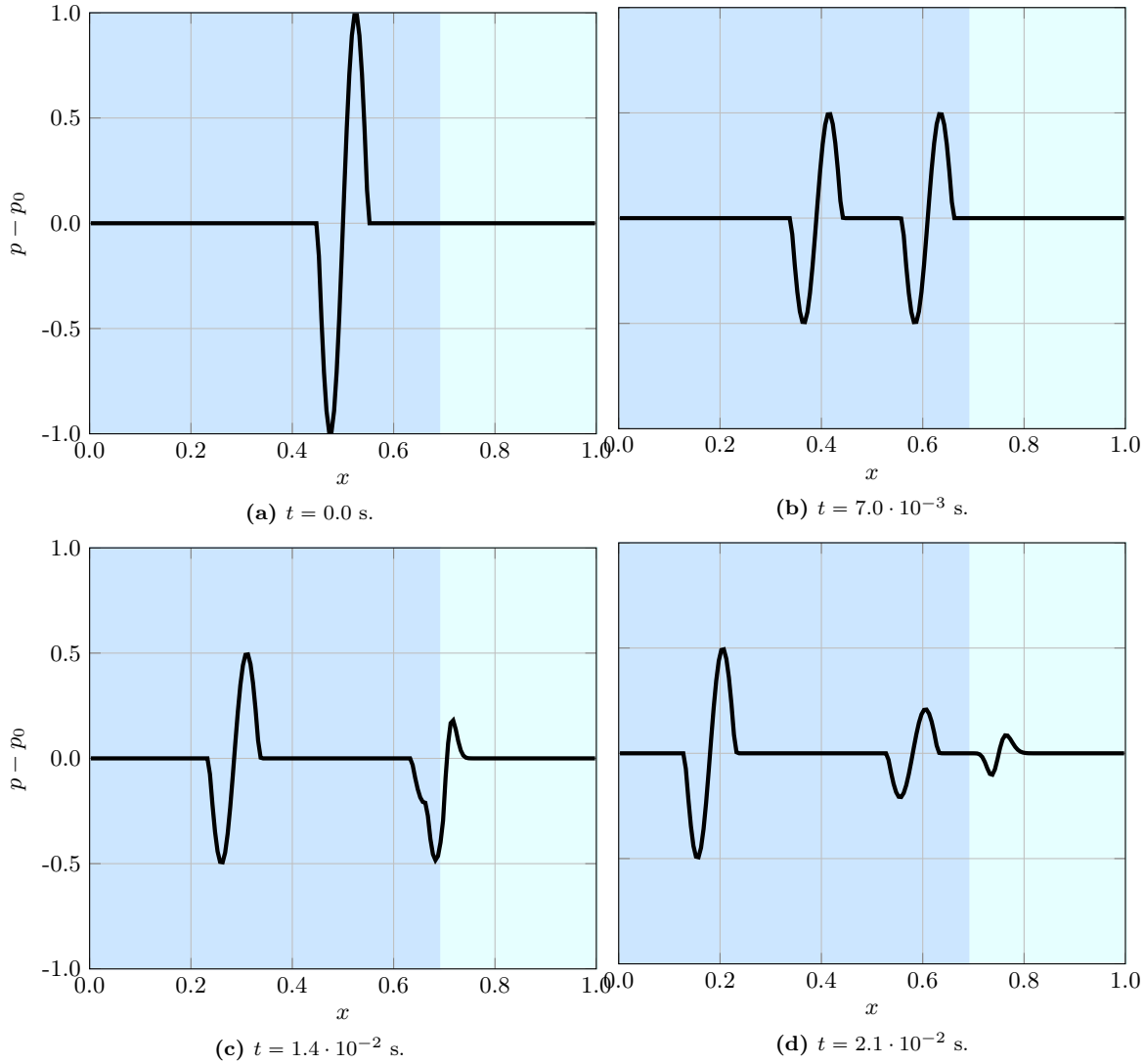
In Figure 5.9c, $\epsilon \gg 1$ and the wave of the 4-equation system is fully transmitted, propagates at c_{Frozen} velocity and is little attenuated in the mixture zone. This behavior is also recovered in Figure 5.9g, where $\epsilon \gg 1$ and $\nu \gg 1$. This regime corresponds to the frozen regime.

In Figure 5.9f, the values of ϵ and ν are such that the resonance regime is reached. There, we can observe a strong deformation of the wave when it crosses the two zones interface.

Finally in Figure 5.9h, we show a case where the modified scheme of the previous section becomes unstable: one can see indeed high-frequency pressure oscillations in the mixture zone (in the red circle).

5.3.2.2 Numerical dispersion relations

From the test case with initial conditions (5.27), we compute the numerical dispersion relations and compare them with the analytical ones. For simplicity of computations, these relations are not functions of the acoustic wavelength (as in Chapter 2), but they vary with the values of the parameters: ϵ for the 4-equation model, ϵ and ν for the 5-equation model. This has the drawback however of complicating the



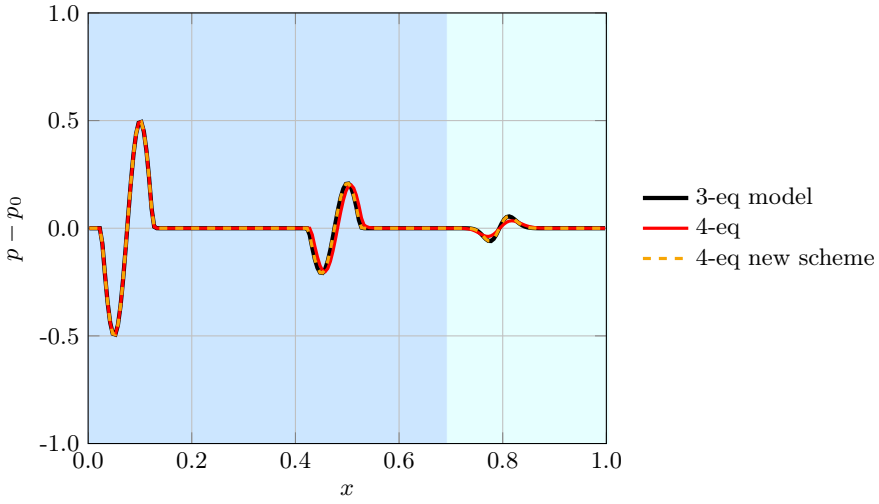
(a) $t = 0.0$ s.

(b) $t = 7.0 \cdot 10^{-3}$ s.

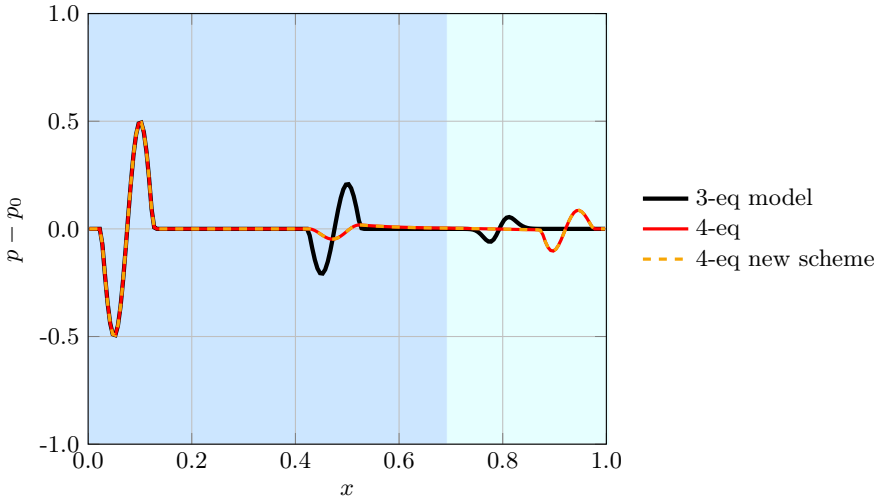
(c) $t = 1.4 \cdot 10^{-2}$ s.

(d) $t = 2.1 \cdot 10^{-2}$ s.

Figure 5.8: *Propagation of an acoustic wave under the 3-equation model from a liquid zone ($\alpha = 1.0 \cdot 10^{-10}$ in the darkest blue zones) to a mixture zone ($\alpha = 1.0 \cdot 10^{-4}$ in the lightest blue zones). Results are presented for different times of the simulation.*

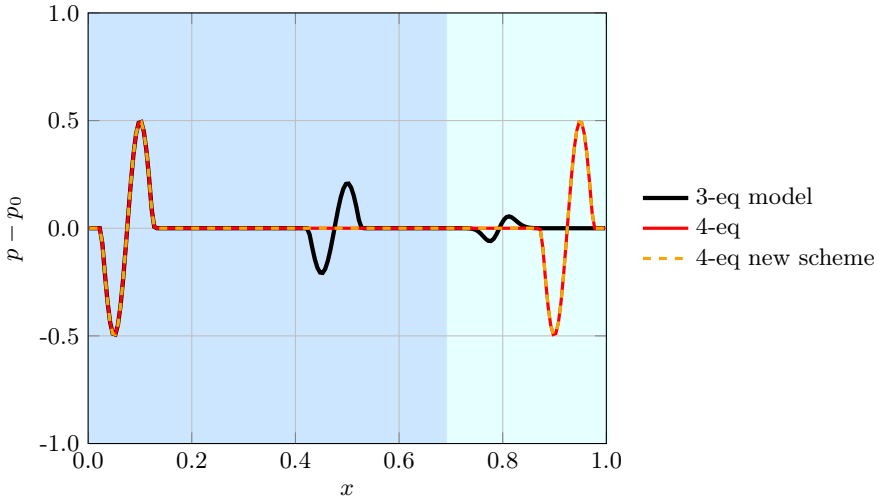


(a) $\epsilon = 1.0 \cdot 10^{-15}$.

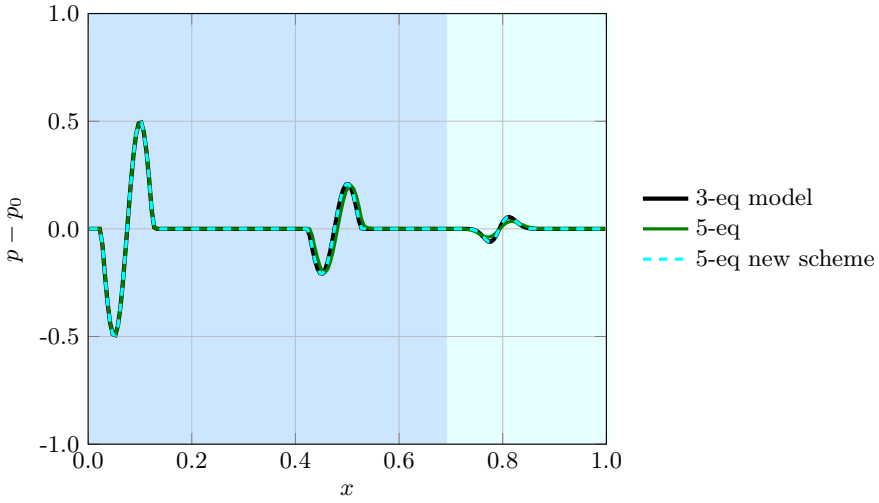


(b) $\epsilon = 1.0 \cdot 10^{5}$.

Figure 5.9: *Propagation of an acoustic wave from a liquid zone ($\alpha = 1.0 \cdot 10^{-10}$ in the darkest blue zones) to a mixture zone ($\alpha = 1.0 \cdot 10^{-4}$ in the lightest blue zones). Results for the three models and for different values of ϵ and ν are presented at time $t_f = 2.8 \cdot 10^{-4}$ s.*

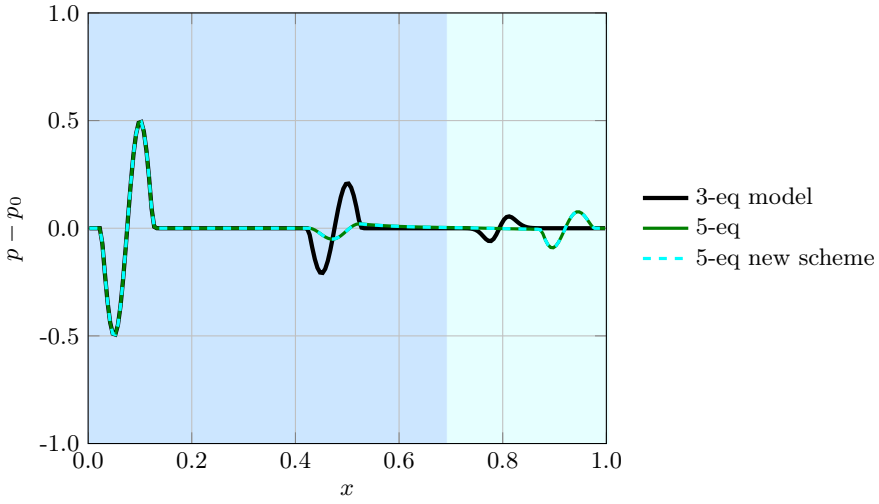


(c) $\epsilon = 1.0 \cdot 10^{10}$.

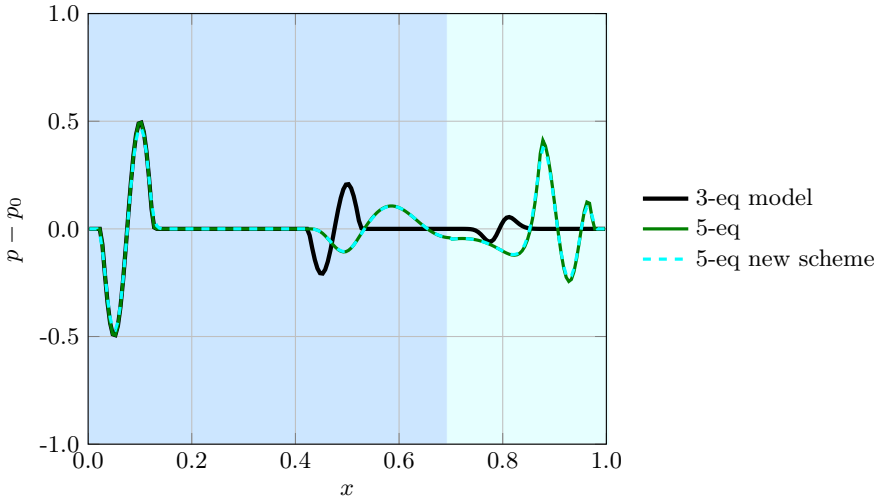


(d) $\epsilon = 1.0 \cdot 10^{-6}$, $\nu = 1.0 \cdot 10^{-15}$.

Figure 5.9: *Propagation of an acoustic wave from a liquid zone ($\alpha = 1.0 \cdot 10^{-10}$ in the darkest blue zones) to a mixture zone ($\alpha = 1.0 \cdot 10^{-4}$ in the lightest blue zones). Results for the three models and for different values of ϵ and ν are presented at time $t_f = 2.8 \cdot 10^{-4}$ s.*

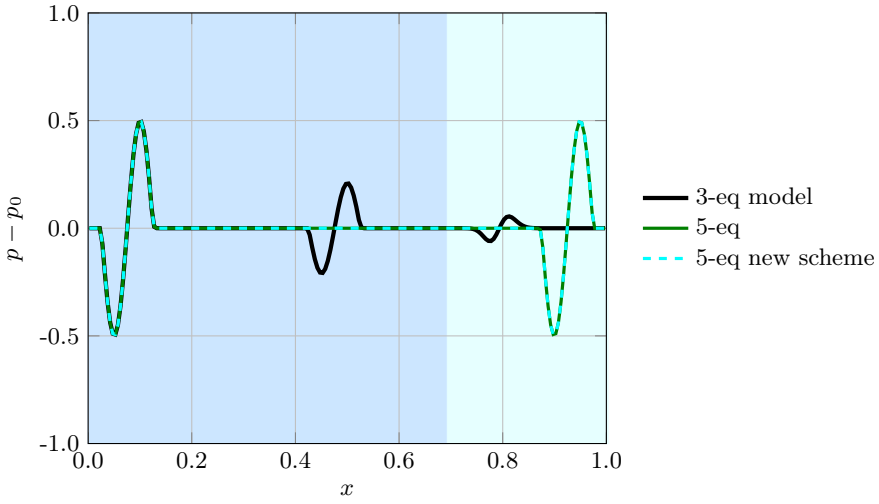


(e) $\epsilon = 1.0 \cdot 10^{5}$, $\nu = 1.0 \cdot 10^{-15}$.

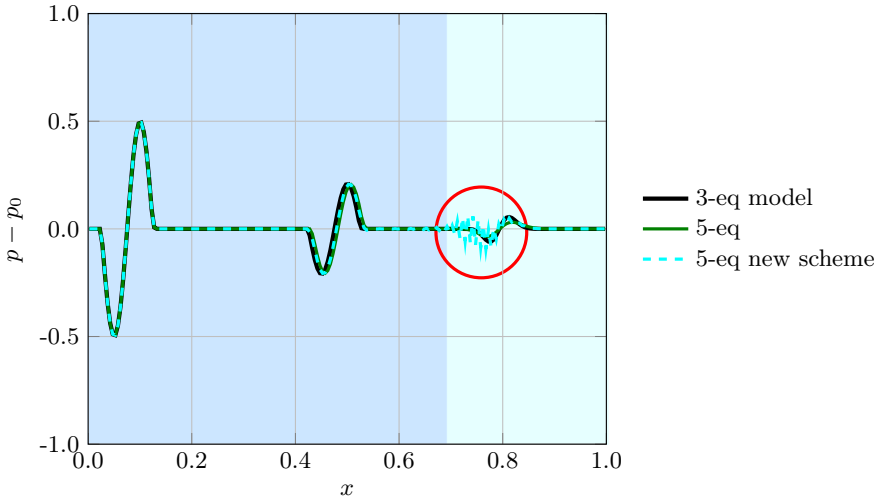


(f) $\epsilon = 1.0 \cdot 10^{4}$, $\nu = 1.0$.

Figure 5.9: *Propagation of an acoustic wave from a liquid zone ($\alpha = 1.0 \cdot 10^{-10}$ in the darkest blue zones) to a mixture zone ($\alpha = 1.0 \cdot 10^{-4}$ in the lightest blue zones). Results for the three models and for different values of ϵ and ν are presented at time $t_f = 2.8 \cdot 10^{-4}$ s.*



(g) $\epsilon = 1.0 \cdot 10^{10}$, $\nu = 1.0 \cdot 10^{10}$.



(h) $\epsilon = 4.8 \cdot 10^{3}$, $\nu = 1.0 \cdot 10^{-2}$.

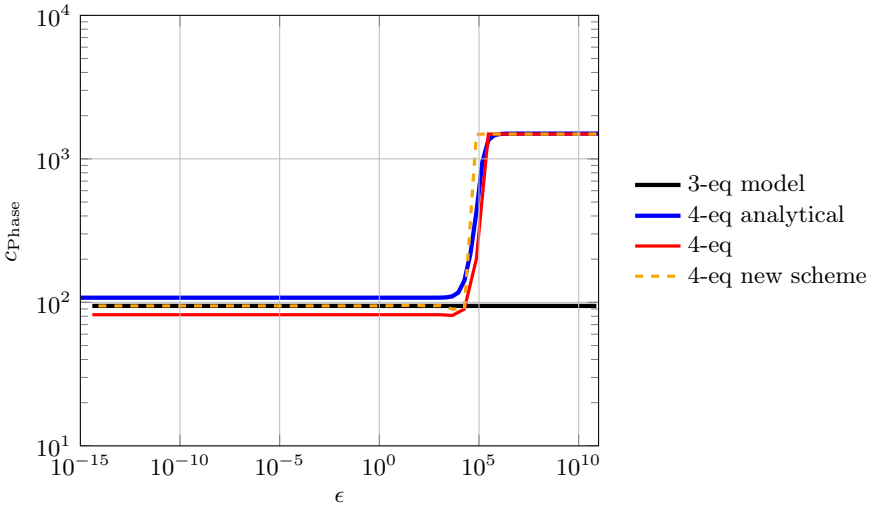
Figure 5.9: *Propagation of an acoustic wave from a liquid zone ($\alpha = 1.0 \cdot 10^{-10}$ in the darkest blue zones) to a mixture zone ($\alpha = 1.0 \cdot 10^{-4}$ in the lightest blue zones). Results for the three models and for different values of ϵ and ν are presented at time $t_f = 2.8 \cdot 10^{-4}$ s.*

analytical formulae for the 5-equation system, due to the resonance phenomena. For this system, only the asymptotic behaviors ($\epsilon \ll 1$ and $\nu \ll 1$) are studied.

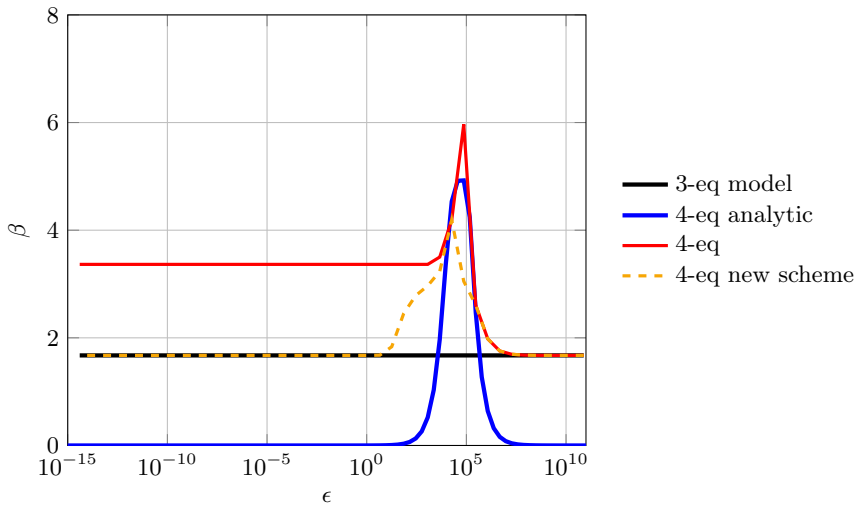
For each simulation, the computation is stopped when the wave reaches a given location x_{stop} in the domain. From the solution at this time step t_{stop}, the phase velocity is given by $(x_{\text{stop}} - x_{\text{init}})/t_{\text{stop}}$. The spatial attenuation is given by half the ratio of the wave magnitude between t_{stop} and the initial time.

In Figures 5.10, we show the results for the 3-equation and 4-equation models when ϵ varies. Since the 3-equation model is independent of ϵ, the relations are constant: the phase velocity is c_{Wood}, and the spatial attenuation is a measure of the numerical dissipation. For the 4-equation model, the numerical dispersion relations are globally close to the analytical ones. The transition from c_{Wood} to c_{Frozen} occurs at the same value for ϵ. Moreover, it also corresponds to the maximum spatial attenuation. For small and large values of ϵ, this attenuation should be decreasing. But numerically this attenuation is bounded from below by the numerical dissipation. One can note that the modified scheme from paragraph 5.3.1 gives results that are closer to the 3-equation model for low value of ϵ.

In Figure 5.11, we present the dispersion relations for the 5-equation model for three fixed values of ϵ: $1.0 \cdot 10^{-15}$, $1.0 \cdot 10^{4}$ and $1.0 \cdot 10^{15}$. For the 5-equation model, the analytical dispersion relations are more complex to compute, since in the transition zone, multiple modes are excited. Consequently, we only show the numerical dispersion relations, that correspond to the less attenuated modes. We can see that for $\epsilon = 1.0 \cdot 10^{-15}$, the results tend to the dispersion of the 3-equation model for low values of ν. For $\epsilon = 1.0 \cdot 10^{4}$, the 4-equation phase velocity and spatial attenuation are recovered for low values of ν. Finally, for $\epsilon = 1.0 \cdot 10^{15}$, the system behaves the same whatever the value of ν. Once more, for low values of ϵ and ν, the correction proposed in paragraph 5.3.1 enables to recover the same attenuation rate as the 3-equation system. However, in the case of the 5-equation model, the analytical phase velocity is more complex to estimate and in some cases, the correction underestimates the real velocity of information, which raises instabilities.

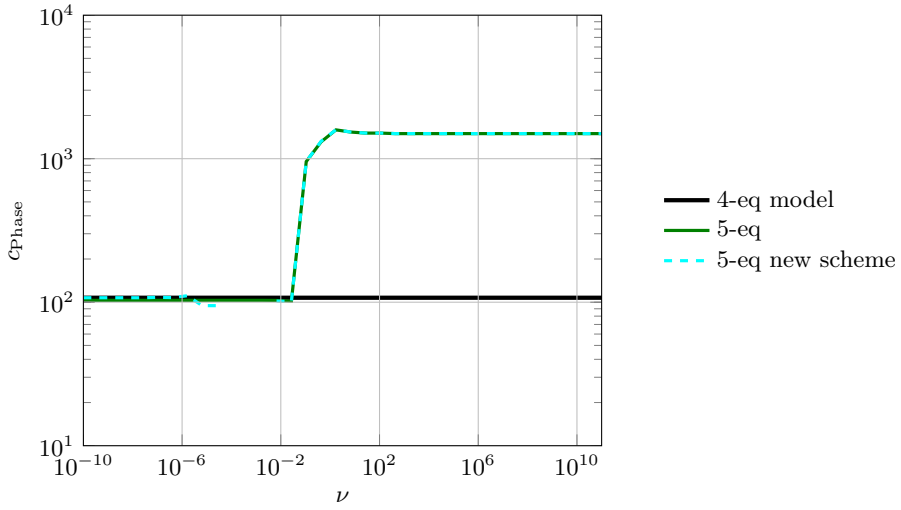


(a) Phase velocity according to ϵ.

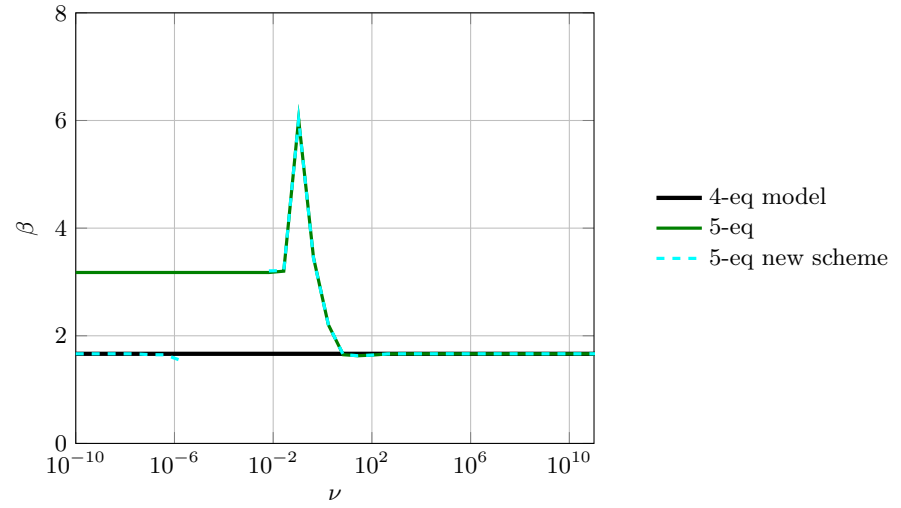


(b) Spatial attenuation according to ϵ.

Figure 5.10: *Numerical dispersion relations for the 3-equation (black curve) and 4-equation (red curve for the Relax-B scheme and yellow curve for correction presented in 5.3.1) models, computed from the test of an acoustic wave propagation ($\lambda = 0.1$ m) in a gas-liquid mixture with $\alpha = 1.0 \cdot 10^{-2}$. The analytical solution is given in blue dashed line.*

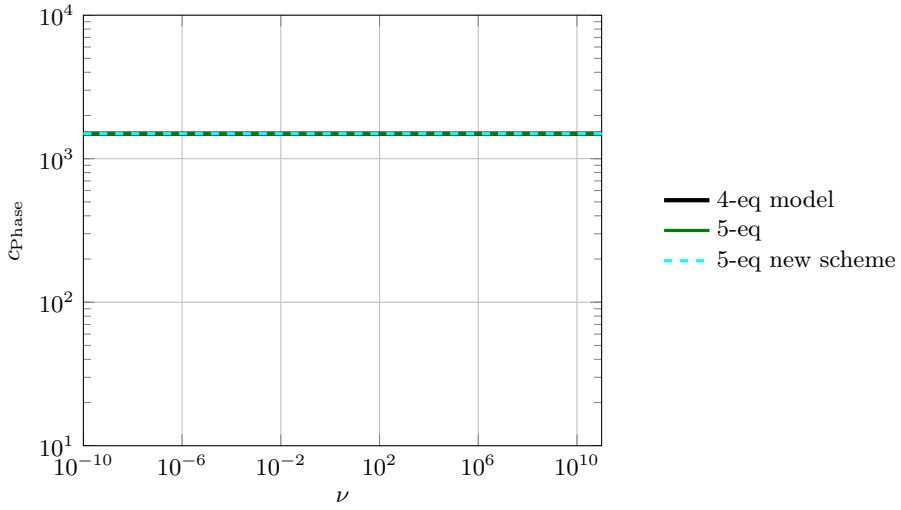


(a) Phase velocity according to ν, $\epsilon = 1.0 \cdot 10^{-15}$.

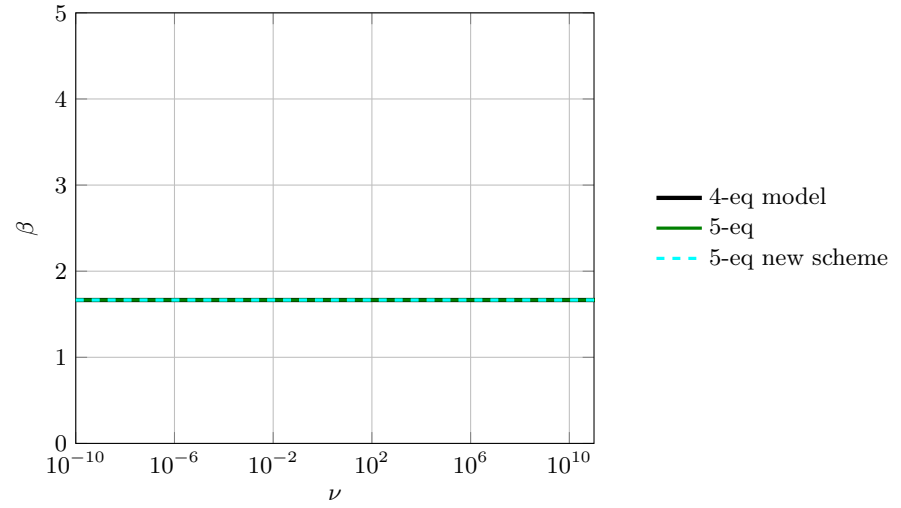


(b) Spatial attenuation according to ν, $\epsilon = 1.0 \cdot 10^{-15}$.

Figure 5.11: *Numerical dispersion relations for the 3-equation or 4-equation (black curve) and 5-equation models (green curve for the Relax-B scheme and cyan curve for the modified one), computed from the test of an acoustic wave propagation ($\lambda = 0.1$ m) in a gas-liquid mixture with $\alpha = 1.0 \cdot 10^{-2}$.*

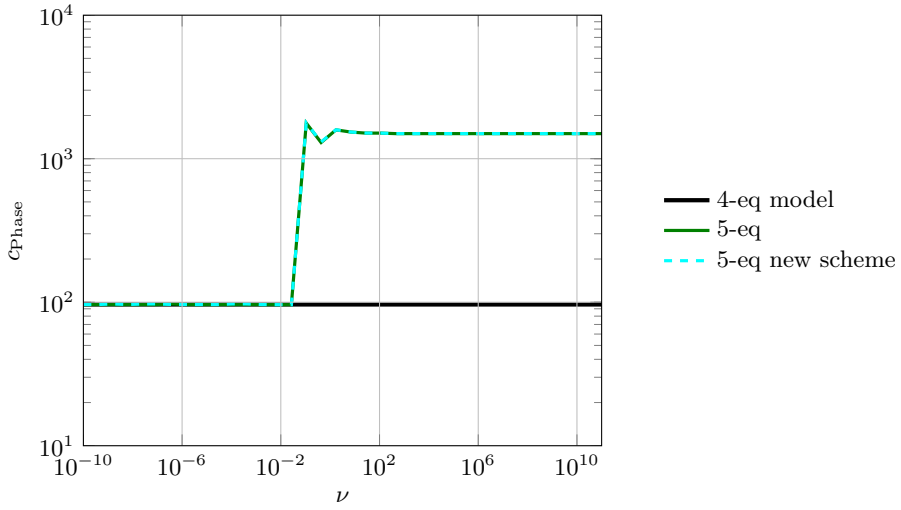


(c) Phase velocity according to ν, $\epsilon = 1.0 \cdot 10^{15}$.

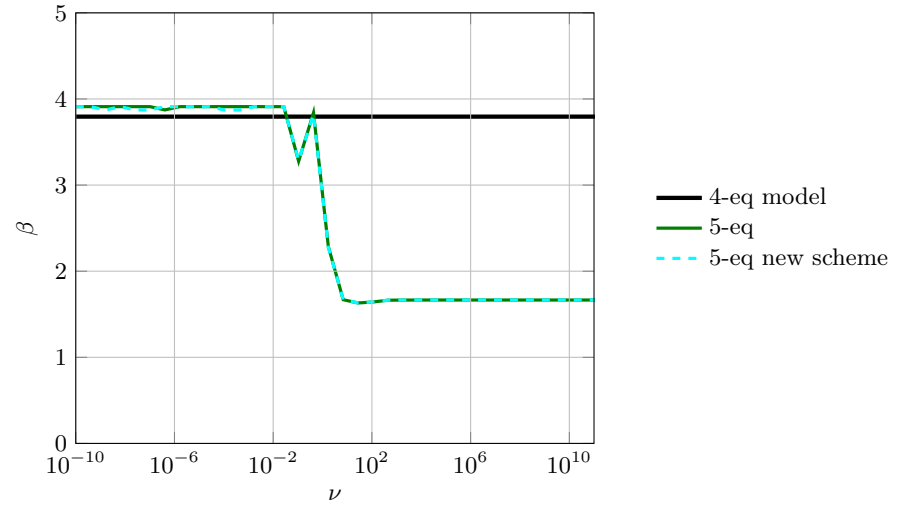


(d) Spatial attenuation according to ν, $\epsilon = 1.0 \cdot 10^{15}$.

Figure 5.11: *(continued) Numerical dispersion relations for the 3-equation or 4-equation (black curve) and 5-equation models (green curve for the Relax-B scheme and cyan curve for the modified one), computed from the test of an acoustic wave propagation ($\lambda = 0.1$ m) in a gas-liquid mixture with $\alpha = 1.0 \cdot 10^{-2}$.*



(e) Phase velocity according to ν, $\epsilon = 1.0 \cdot 10^4$.



(f) Spatial attenuation according to ν, $\epsilon = 1.0 \cdot 10^4$.

Figure 5.11: *(continued) Numerical dispersion relations for the 3-equation or 4-equation (black curve) and 5-equation models (green curve for the Relax-B scheme and cyan curve for the modified one), computed from the test of an acoustic wave propagation ($\lambda = 0.1$ m) in a gas-liquid mixture with $\alpha = 1.0 \cdot 10^{-2}$.*

5.4 Conclusion

Let us summarize the different aspects that have been addressed in this chapter. We have first studied methods for integrating the source terms of our two-phase systems of equations. The difficulty of these integrations lie in the fact that the ODE may become very stiff in the neighborhood of $\alpha \in \{0, 1\}$, and high-order A-stable and L-stable solvers, such as Radau5 (Hairer and Wanner (1996)) or VODE (Brown, Byrne, and Hindmarsh (1989b)) fail in the integration of the source term of the 5-equation system. This has motivated the development of solvers dedicated to these particular source terms. The solutions which have been retained are based on the exact solutions of an approximate ODE and an additional clipping which ensures $\alpha \in]0, 1[$. In the future, more work can be done in order to:

- estimate the order of accuracy of these approximate solutions,
- further analyze the origins of the stiffness of the ODE,
- improve the accuracy of the methods, considering high-order integration methods with A-stable and L-stable properties and which ensure $\alpha \in]0, 1[$ whatever the time of integration Δt.

Then we have tested the whole numerical strategy, which is based on operator splitting, on Relax-B scheme and on the approximate source term integration. With a last modification of Relax-B, we have been able to recover numerically a form of the analytical dispersion relations of Chapter 2 and correct asymptotic behaviors of the numerical solution. Here again, further investigation of some aspects seem necessary:

- we have hit a problem of large numerical diffusion in the limit $\epsilon \to 0$ for the 4-equation and 5-equation models. A first cure has been proposed, which is based on the modification of the characteristic waves in the Riemann problem. However, this modified scheme seems to be unstable in some configurations, and may probably loose its accuracy when the discretization is refined. A better modified scheme should be proposed,
- higher-order ODE solvers could be used and the same numerical tests could be performed and compared with the results presented here, so as to estimate the impact of the order of accuracy of the source term integration.

Chapter 6

Accurate numerical strategies on quadrangular non-conforming meshes

6.1 Introduction

The general framework of the numerical simulations we want to perform involves the use of AMR techniques, and more specifically cell-based Adaptive Mesh Refinement (AMR), by using the p4estlibrary (presented in Chapter 7) and its applicative code CanoP(Chapter 8). We propose in this chapter a set of methods which enable to benefit from this AMR framework. In particular, three points are studied here:

- The cell-based AMR method generates *non-conforming* grids. The numerical solutions of the two-phase models must then be computed on these particular grids, which can be considered either as an unstructured meshes made of cells with a varying number of faces, or as cartesian-like grids, provided that all cells have a quadrangular shape in 2D (and hexahedral shape in 3D).
- One must also choose a strategy for mesh adaptation, so as to ensure a constant accuracy of the numerical solution while reducing the cost of the resolution in specified spatial zones.
- Finally, following numerical experiments, it has been noticed that the resolution of compressible two-phase system on non-conforming grids in the low-Mach number limit gives very poor results. For monophasic flows, this problem is well-known and different cures to the low-Mach issue exist. One of them was also proposed for two-phase flows (Girardin (2014)), but is had never been tested in an AMR framework.

These three aspects are developed in this chapter and we present the methods which have been implemented in the CanoPcode. But first, let us recal two notions that will be referred to at different stages of this chapter.

Non-conforming grids *Cartesian non-conforming* meshes are composed of cells of the same topology type (hexahedron, tetrahedron...) but of possibly different sizes. Contrary to cartesian meshes and usual unstructured meshes, in non-conforming meshes, although the number of faces of a cell is fixed, the number of its neighbors across the faces may vary. If the non-conforming mesh is of cartesian type, this means that its cells can be given an order, linking each cell to its neighbors. A non-conforming cartesian mesh can be built departing from a conforming cartesian one: cells are then recursively split into new cartesian sub-grids.

It is standard to give a limit to the number of neighbors across a face, and thus the size ratio of two neighboring cells. When this ratio is at most two, the mesh is said to be **2:1 balanced**. If it is built from a coarse cartesian mesh, that is to be locally refined, the 2:1 balance constraints each cell to be split into at most 4 new cells in 2D (and 8 new cells in 3D). Moreover, when a recursive refinement process is used, the last step must consist in verifying the 2:1 balance constraint between all neighboring cells (an

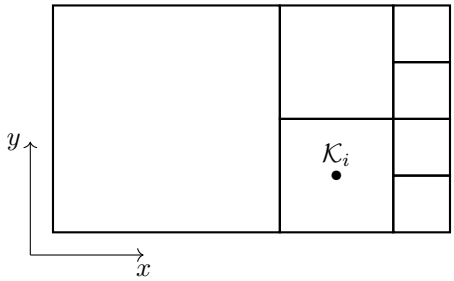


Figure 6.1: *Example of a non-conforming grid.*

illustration of non-conforming mesh with 2:1 balance is given in Figure 6.1). Let us note that the number of newly created cells in case of refinement is not necessary 4 (or 8) but can be less. Such anisotropic grids can be found in the works of Freret and Groth (2015) for example.

In the following, we adopt classic notations pertaining to unstructured meshes to describe the AMR grid: the cell i is noted K_i, its volume is $|K_i|$ while $|\Gamma_{ij}|$ and $\mathbf{n}_{ij}$ are respectively the surface and the unit normal of the interface between two neighboring cells i and j (or **sub-face** $|\Gamma_{ij}|$). The vector $\mathbf{n}_{ij}$ is oriented from cell i to cell j. We note $\mathcal{N}(i)$, the set of all neighboring cells of cell i and $\mathcal{N}_q(i)$, the set of neighboring cells of cell i in the direction $q = x, y, z$.

Conservative / non-conservative variables In this chapter, we consider hyperbolic system of conservation laws writing:

$$\partial_t \mathbf{W} + \partial_x \mathcal{F}_x(\mathbf{W}) + \partial_y \mathcal{F}_y(\mathbf{W}) + \partial_z \mathcal{F}_z(\mathbf{W}) = \mathcal{S}(\mathbf{W}), \tag{6.1}$$

where $\mathbf{W}$ is the vector of conservative variables, $\mathcal{S}$ stands for body force source terms and the fluxes $\mathcal{F}_q$ verify the rotational invariance property $\mathcal{F}_q(\mathbf{W}) = R_q^{-1}\mathcal{F}_x(R_q\mathbf{W})$, with R_q being defined as the associated rotation matrix.

While conservative variables are useful when using Godunov methods (as presented in Chapter 4), other sets of variables may also be of interest. For example, when using a second-order MUSCL scheme (as presented later), it is recommended to use the *primitive* variables or *entropic* ones, especially during the slope limiting process. For the Euler system of equations, conservative variables are $(\rho, \rho\mathbf{u}, \rho E)^T$, primitive ones are $(\rho, \mathbf{u}, E)^T$, *entropic* variables $(\rho, \mathbf{u}, s)^T$ or even $(p, \mathbf{u}, s)^T$ (cf Berthon (2005)).

Ω denotes the open convex set of admissible states for the conservative variables, meaning that any $\mathbf{W} \in \Omega$ must satisfy some restrictions for the solution to keep a physical meaning. For the Euler equations, Ω is convex and defined as the set of positive density and internal energy:

$$\Omega = \left\{ \mathbf{W} \in \mathbb{R}^N \quad | \quad \rho > 0 \quad \text{and} \quad E - |\mathbf{u}|^2/2 > 0 \right\} \tag{6.2}$$

Remark 25. *We expect other variables to satisfy such admissibility conditions, such as pressure p or volume and mass fractions in the case of two-phase flows. These conditions are ensured by the state laws associated with the model.*

6.2 Finite volume methods applied to non-conforming grids

In the following paragraphs, we present a series of numerical features, which are related to the resolution of conservative systems of equations such as (6.1), using finite volume methods on non-conforming grids. These features are used in the CanoPcode (presented in Chapter 8).

6.2.1 Flux computation and solution update

Let us consider a cell K_i of a 2D non-conforming cartesian grid. This cell has four faces, but may have more than four neighbors. The strategy we choose to update the solution in cell K_i is to compute

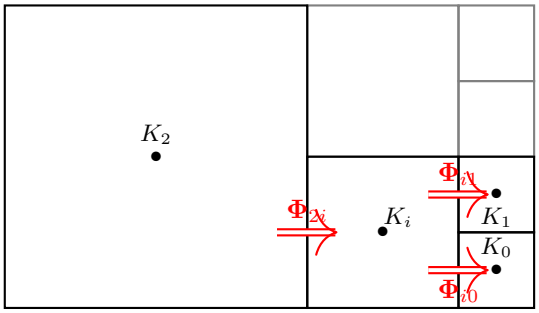


Figure 6.2: *Non-conforming grid and the different fluxes associated with cell K_i in one space direction.*

the numerical fluxes $\mathbf{\Phi}_{ij}$ through each sub-face Γ_{ij} of K_i (as illustrated in Figure 6.2), and update the solution in K_i with the formula:

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n - \frac{\Delta t}{|K_i|} \sum_{j \in \mathcal{N}(i)} |\Gamma_{ij}| \mathbf{\Phi}_{ij}^T \mathbf{n}_{ij},$$

where Δt is the time step of integration, whose value is discussed below. To ensure the conservation property, the solution in the neighboring cells of K_i will be updated using the same values of numerical fluxes $\mathbf{\Phi}_{ij}$.

Remark 26. *The choice of computing the fluxes across each sub-face is especially adapted to the cell-based AMR approach, where non-conforming faces may be encountered for a lot of cells. In block-based AMR, however, the sub-grids are made of several cells: each sub-grid is considered as a whole sub-domain and its borders are considered as boundary zones that necessitate special treatment. Usually, a layer of ghost cells with the same level of refinement as inside the block is created at this boundary zone. The cells in the ghost layer are filled with copied, averaged or interpolated values of the solution of a coarser level grid (in case of patch-based), or of the neighboring patch or block. The numerical fluxes are then always computed between cells with the same level of refinement.*

The time step is computed using a CFL condition over the whole grid:

$$\Delta t \leqslant \frac{1}{2} \min_{i, j \in \mathcal{N}(i)} \left(\frac{\sigma_{ij} |\Gamma_{ij}|}{|K_i|} \right), \tag{6.3}$$

where σ_{ij} is the maximum wave velocity in the Riemann problem at face Γ_{ij}. The solution is thus integrated in all cells with the same time step. Local time stepping (or *subcycling*) is often associated with AMR techniques (see Berger and Oliger (1984); Berger and Collela (1989); Golay, Ersoy, Yushchenko, and Sous (2015) among others), but has not been tried yet in our work (except first tests in Fikl (2014)).

6.2.2 Dimensional splitting

The structure of the grid which is considered here (Cartesian-like grids) enables to use standard dimensional splitting (Leveque (2004)). This splitting consists in updating the solution in successive steps using the numerical flux operators associated with the dimension components of the physical flux $\mathcal{F}_x$, $\mathcal{F}_y$ and $\mathcal{F}_z$ from system (6.1). In the following, we will not consider the source terms $\mathcal{S}$: they are systematically solved using operator splitting (see Chapter 5). A consequence of this choice is discussed in Appendix D when a gravity source term is added to the equations and well-balanced issues arise.

In the following, we present three dimensional splittings:

- an *unsplit* method, where fluxes are computed in all directions, and the solution update is performed in one iteration,
- a Godunov (or Lie) splitting method, where each direction is updated successively, necessitating thus two iterations in 2D (and three in 3D),

- a Strang splitting method, where each direction is updated over a half time step successively given an specified order, and then are updated in another half time step successively in the reverse order.

Remark 27. *These three methods have been integrated in CanoP, since some numerical schemes (like the all-regime scheme presented in section 6.5), necessitate an unsplit strategy. For other cases, the Godunov and Strang splitting give more accurate solutions, for the reasons presented below.*

Unsplit scheme The full scheme using the unsplit method for updating the convective part of (6.1) in cell K_i from time t^n to time $t^{n+1} = t^n + \Delta t$ is:

$$\mathbf{W}_i^{n+1} = \left(\square_x^{\Delta t} + \square_y^{\Delta t} + \square_z^{\Delta t}\right) \mathbf{W}_i^n - 2\mathbf{W}_i^n, \tag{6.4}$$

where the operator $\square_q^{\Delta t}$ is defined by

$$\square_q^{\Delta t}\mathbf{W}_i = \mathbf{W}_i - \frac{\Delta t}{|K_i|} \sum_{j \in \mathcal{N}_q(i)} |\Gamma_{ij}| (\mathbf{e}_q^T \mathbf{n}_{ij}) R_q^{-1} \Phi_{ij}, \quad \Phi_{ij} = \Phi(R_q \mathbf{W}_i, R_q \mathbf{W}_j), \tag{6.5}$$

for $q = x, y, z$ and $(\mathbf{W}_L, \mathbf{W}_R) \mapsto \Phi$ being a choice of numerical flux.

The time step Δt is chosen to satisfy the CFL-like condition (6.3). This restriction ensures the stability of the solution, as explained is Chapter 4. It can also be seen in the following way: *information* should not propagate through more than one cell during the time step Δt. Moreover, since all directions are solved and since information may propagate transversally to the directions, condition (6.3) ensures that the information propagation does not reach a corner neighbor during Δt.

Dimensional splitting: In order to approximate the solutions of (6.1), dimensional splitting consists, during a time step Δt, in solving successive one-dimensional problems, associated with each direction, using a discretization of (6.1) with a classic 1D finite volume method.

The particularity of non-conforming meshes is that sub-faces have to be dealt with. However, when the non-conforming grid is cartesian-like, at each sweep in the direction, one can treat all faces and sub-faces orthogonal to this direction, whatever their number.

The full scheme for advancing the convective part of (6.1) in cell K_i from time t^n to time t^{n+1} is then given for a first order dimensional slitting by:

$$\mathbf{W}_i^* = \square_x^{\Delta t}\mathbf{W}_i^n, \qquad \mathbf{W}_i^{**} = \square_y^{\Delta t}\mathbf{W}_i^*, \qquad \mathbf{W}_i^{n+1} = \square_z^{\Delta t}\mathbf{W}_i^{**},$$

or which can also be written:

$$\mathbf{W}_i^{n+1} = \square_z^{\Delta}\square_y^{\Delta}\square_x^{\Delta}\mathbf{W}_i^n. \tag{6.6}$$

Due to the dimensional splitting, the CFL constraint is not as restrictive as for unsplit methods: since we solve consecuting 1D problems, the CFL condition may become:

$$\Delta t \leqslant \min_q \min_{i,j \in \mathcal{N}_q(i)} \left(\frac{\sigma_{ij} |\Gamma_{ij}|}{|K_i|} \right), \quad q = x, y, z.$$

We can note that, compared with formula (6.3), the Courant number $\frac{1}{2}$ was changed to 1: although stability analysis of 1D schemes often requires a Courant number $C = \frac{1}{2}$ (Godlewski and Raviart (1996)), in practice, choosing $C = 1$ gives stable results. One advantage of Lie splitting over unsplit method is that time steps can be twice as large. For some finite volume schemes, this can have an influence on the numerical diffusion of the scheme, and consequently on the accuracy of the numerical solution (as noticed in Leveque (2004)).

However, as mentioned in the introduction of Part II Lie operator splitting introduces a first order splitting error in time, unless the operators $\square_q^{\Delta t}$ commute. To ensure a higher-order method, one can then increase the order of accuracy of the solution by using a Strang splitting, or by alternating the order in which the directions are solved, from one time iteration to the other, as proposed in Leveque (2004).

Strang splitting: Strang splitting is used here, as Lie splitting, by treating all at once the faces and sub-faces orthogonal to successive directions. In the case of system (6.1), the procedure comes down to solve the equations according to the x then y, then z directions on a half time step, and again z then y then x directions on the other half time step. The update procedure (6.6) is then replaced by:

$$\mathbf{W}_i^{n+1} = \square_x^{\Delta t/2}\square_y^{\Delta t/2}\square_z^{\Delta t/2}\square_z^{\Delta t/2}\square_y^{\Delta t/2}\square_x^{\Delta t/2}\mathbf{W}_i^n, \tag{6.7}$$

leading to a numerical scheme that is second order in time and space, provided that each operator $\square_q^{\Delta t/2}$ is itself at least second order accurate in time and space. One can note that the z direction is integrated over two half time steps, instead of one entire. The reason for this is that we consider, in our works, numerical fluxes, whose order of accuracy is at most two. Then, there can be a difference between integrating successively two times over $\frac{1}{2}\Delta t$ and one time over Δt. In order to keep an isotropic level of accuracy, we have chosen to integrate the z direction the way of (6.7). It was noticed in Leveque (2004) that Strang splitting is not always more advantageous than Godunov splitting, especially for schemes with a diffusion increasing when the number of time iterations is increased.

Remark 28. *When source terms operators $\square_S$ are considered, the formula (6.7) can be modified as:*

$$\mathbf{W}_i^{n+1} = \square_x^{\Delta t/2}\square_y^{\Delta t/2}\square_z^{\Delta t/2}\square_S^{\Delta t}\square_z^{\Delta t/2}\square_y^{\Delta t/2}\square_x^{\Delta t/2}\mathbf{W}_i^n,$$

if the source term is the stiffest operator (and assuming it is solved over Δt with a sufficient accuracy).

Conclusion on dimensional splittings: We have presented three methods for the integration of the dimensional components of the flux operator: one unsplit method, which is not the most accurate, but is required for some numerical schemes (the all-regime scheme for example), when dealing with deformed grids, or with triangular grids (see end of Chapter 8) ; the Lie splitting method, which gives a priori a first-order splitting method and a Strang splitting method, which is higher-order accurate, but requires to integrate over half time steps.

6.2.3 Boundary conditions

Let us give here some details on the boundary conditions that can be applied at the domain borders, and more particularly on the three conditions that will be used in the numerical tests of Chapter 9, namely the periodic, *infinite domain* conditions and symmetric conditions.

Periodic boundary conditions: With these conditions, the domain is assumed to be cyclic or periodic. For a quadrangular domain, this condition assumes that the cells at the right border of the domain are neighboring the cells of the left border, and that the cells at the upper border are neighbors of the cells at the lower border. The treatment of this boundary condition requires particular attention with the grid management (when defining the connectivities of the cells), but the numerical scheme is not impacted by this boundary condition.

Infinite domain: The infinite domain let all waves traveling outside of the domain, and assumes that no new quantity enters the domain. Let $\mathbf{W} = (\mathbf{u}, \mathbf{V})^T$ be the vector of variables composing the solution, with $\mathbf{u}$ the material velocity and $\mathbf{V}$ the other variables. If $\mathbf{W}_i$ is the value of the solution in cell K_i at the domain border, with face Γ_{ij} being the domain boundary with normal $\mathbf{n}_{ij}$ oriented along the q direction, then the numerical flux at this face is given by $\mathbf{\Phi}_{ij} = \mathbf{\Phi}(R_q\mathbf{W}_i, R_q\mathbf{W}_i)$.

Symmetric boundary conditions: Let us consider the same notations as in the previous paragraph. With the symmetric boundary conditions, no matter is allowed to go out of the domain. This is ensured by setting $\mathbf{W}_j = (\mathbf{u} - 2(\mathbf{u}^T\mathbf{n}_{ij})\mathbf{n}_{ij}, \mathbf{V})^T$ and computing the numerical flux at the boundary with $\mathbf{\Phi}_{ij} = \mathbf{\Phi}(R_q\mathbf{W}_i, R_q\mathbf{W}_j)$.

Remark 29. *The last two conditions are defined in a classic way (Leveque (2004)) whatever the type of grid. Using non-conforming grids is not an issue here.*

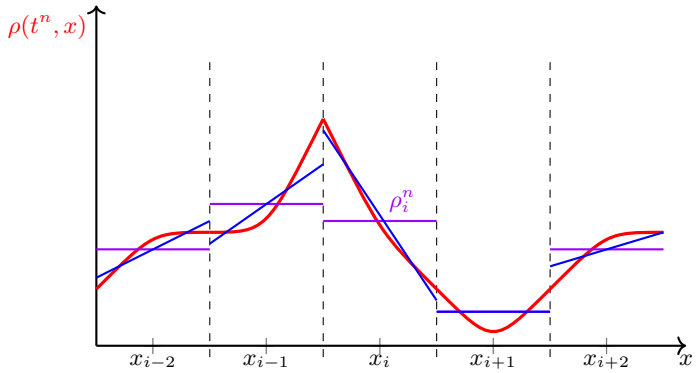


Figure 6.3: *Illustration of piecewise linear reconstruction (in blue) of a scalar function in the finite volume framework.*

6.3 MUSCL-Hancock second order reconstruction

Here, we consider again cartesian-like non-conforming grids and we propose to use the dimensional splitting strategy presented above to solve the solution through a second-order scheme in each direction. Spatial reconstructions of the solution are made according to the MUSCL scheme (van Leer (1979)) and with slope limiters (Sweby (1984)), and the second order in time is provided by the predictor-corrector Hancock's method (van Leer (1984)).

6.3.1 1D slope reconstruction

A way to obtain a second-order extension of the Godunov method is to consider piecewise linear solutions, instead of piecewise constant ones. This method was first proposed in van Leer (1979) and is named Monotonic Upstream Schemes for Conservation Laws (MUSCL).

In fact, these piecewise linear solutions are *reconstructed* at each time step, and then used for updating the piecewise constant solutions as defined at the beginning of Chapter 4. In each cell $\mathcal{K}_i$, it is then evaluated using the local mean value $\mathbf{W}_i$, and the values in the neighboring cells $\mathbf{W}_{i-1}$ and $\mathbf{W}_{i+1}$ (see illustration in Figure 6.3). This helps computing the values $\mathbf{W}_{i,i-\frac{1}{2}}$ and $\mathbf{W}_{i,i+\frac{1}{2}}$ at the interfaces of cell $\mathcal{K}_i$ in a more accurate way than with the piecewise constant approach. The idea is, afterwards, to solve the Riemann's problems using the reconstructed values $\mathbf{W}_{i-1,i-\frac{1}{2}}$ and $\mathbf{W}_{i,i-\frac{1}{2}}$ at interface $i-\frac{1}{2}$ and values $\mathbf{W}_{i,i+\frac{1}{2}}$ and $\mathbf{W}_{i+1,i+\frac{1}{2}}$ at interface $i+\frac{1}{2}$. The linear reconstruction does not necessarily apply on the conservative variables: several authors (Berthon (2005) for example) recommend to choose the primitive or *entropic* variables instead.

Then let us consider a smooth change of variables Λ and a set of new variables $\mathbf{U}$ such that $\mathbf{W} = \Lambda(\mathbf{U})$. Now, let σ_i be the local slope of the solution in cell $\mathcal{K}_i$. The local linear approximation is defined by:

$$\mathbf{U}(x) = \mathbf{U}_i + \sigma_i \left(x - x_i\right).$$

For a cell of size Δx, the approximations of the conservative variables at the interfaces are then given by:

$$\mathbf{W}_{i,i\pm\frac{1}{2}} = \Lambda\left(\mathbf{U}_i \pm \frac{1}{2}\sigma_i \Delta x\right). \tag{6.8}$$

Remark 30. *When performing a reconstruction on variables different from the conservative ones, one must ensure that the interfacial values are still in the domain of admissible states (meaning ensuring that density ρ stays positive for instance).*

Now, let us note that we have at our disposal three local values (namely $\mathbf{U}_{i-1}$, $\mathbf{U}_i$ and $\mathbf{U}_{i+1}$) to compute a linear function, that only necessitates two of them. Multiple choices can be considered, among them:

- a centered reconstruction for each component U of $\mathbf{U}$:

$$\sigma_{U,i}^{\text{centered}} = \frac{U_{i+1} - U_{i-1}}{x_{i+1} - x_{i-1}},$$

- left and right reconstructions:

$$\sigma_{U,i}^{\text{left,right}} = \frac{U_i - U_{i\pm1}}{x_i - x_{i\pm1}},$$

  where the left slope is assigned $+$ and the right $-$ subscripts.

However, these reconstructions do not always give TVD schemes (see Leveque (2004) for more details). We call a **slope limiter**, a function that defines the local linear reconstruction and that ensures the TVD property of this reconstruction. Usual limiters are minmod, monotonized central, osher, superbee... As we will not use all of them in the future, let us just give the formula for minmod:

$$\sigma_{U,i}^{\text{minmod}} = \begin{cases} \sigma_{U,i}^{\text{left}} \max\left(0, \min\left(1, \dfrac{\sigma_{U,i}^{\text{right}}}{\sigma_{U,i}^{\text{left}}}\right)\right) & \text{if} \quad |\sigma_{U,i}^{\text{left}}| > 0 \\ 0 & \text{else} \end{cases} \tag{6.9}$$

This formula indicates that the slope is chosen as the lowest absolute value between the left and the right reconstructed slopes, if they both have the same sign. The final slope is null if the left and right slopes have opposite signs. This choice is quite constraining, since it involves a clipping at extrema, and thus a loss of accuracy in these zones. Some improvements to this defect can be found in other slope limiters, or with the One-Step-Monotonicity-Preserving schemes of Daru and Tenaud (2004).

However, MUSCL reconstruction method with minmod limiter enables the numerical scheme to be of second order in space in zones of smooth variations of the solution, and is TVD. One now needs a method which is second order in time to hope for a full second-order scheme.

6.3.2 MUSCL-Hancock scheme

Principles and properties of the MUSCL-Hancock scheme: The MUSCL-Hancock method was first presented in van Leer (1984). One can also find a description in Toro (2009) and a stability analysis in Berthon (2006). This space and time second-order method is based on a MUSCL linear reconstruction and on a prediction step for the local variables at an intermediate time step $t^{n+\frac{1}{2}} = t^n + \frac{\Delta t}{2}$. The advantage of MUSCL-Hancock method lies in the prediction step that does not need multiple resolutions of interfacial Riemann's problems (as in the case of Runge-Kutta methods for instance): one needs to compute it once using only the local data of cell $\mathcal{K}_i$ and the slope of the linear MUSCL reconstruction σ_i.

While MUSCL-Hancock method has a low computational cost, it requires nevertheless some constraints to ensure the L1-stability of the solution (see explanations in Berthon (2006)):

1. the first order scheme used to compute the numerical fluxes must be L1-stable,
2. not all slope limiters are compatible with this method, but only those satisfying the following criterion, that ensures the preservation of the convex variable space Ω:

$$\kappa \mathbf{U}_i \pm \sigma_i \Delta x \in \Omega, \tag{6.10}$$

3. the time steps Δt must satisfy conditions that are more complex than the usual CFL constraint, as they depend on different quantities computed during the procedure.

The first constraint is satisfied by the approximate Riemann solvers described in Chapter 4.

A method is proposed in Berthon (2006) to cure the third constraint. In practice, we will see that a CFL of $\frac{1}{2}$ is sufficient for the models and schemes we are using.

As far as the second constraint is concerned, classic slope limiters can be adapted so as to satisfy criterion (6.10). However, we can show that the minmod limiter verifies (6.10), and that is why we only consider

this limiter in the following. Indeed, let us assume that initially $\mathbf{W}_{i-1}, \mathbf{W}_i$ and $\mathbf{W}_{i+1} \in \Omega$. Then, in the case when $\sigma_i^{\mathsf{minmod}} = \sigma_i^{\mathsf{left}} \geqslant 0$, and that conservative variables are reconstructed, we have

$$\begin{aligned}\mathbf{W}_i - \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{left}} &= \mathbf{W}_i - \frac{\mathbf{W}_i - \mathbf{W}_{i-1}}{\Delta x_{\mathsf{left}}} \Delta x_{\mathsf{left}} \\ &= \mathbf{W}_{i-1} \in \Omega\end{aligned}$$

$$\begin{aligned}\mathbf{W}_i + \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{right}} &\leqslant \mathbf{W}_i + \sigma_i^{\mathsf{right}} \Delta x_{\mathsf{right}} \\ &\leqslant \mathbf{W}_{i+1}\end{aligned}$$

and $\mathbf{W}_i + \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{right}} \geqslant \mathbf{W}_i$, then, with Ω convex, we have $\mathbf{W}_i + \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{right}} \in \Omega$.
Assuming that $\sigma_i^{\mathsf{minmod}} = \sigma_i^{\mathsf{left}} \leqslant 0$, we still have $\mathbf{W}_i - \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{left}} = \mathbf{W}_{i-1} \in \Omega$, and now:

$$\begin{aligned}\mathbf{W}_i + \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{right}} &\geqslant \mathbf{W}_i + \sigma_i^{\mathsf{right}} \Delta x_{\mathsf{right}} \\ &\geqslant \mathbf{W}_{i+1}\end{aligned}$$

and $\mathbf{W}_i + \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{right}} \leqslant \mathbf{W}_i$, then, with Ω convex, we have $\mathbf{W}_i + \sigma_i^{\mathsf{minmod}} \Delta x_{\mathsf{right}} \in \Omega$.
A similar demonstration can be performed with $\sigma_i^{\mathsf{minmod}} = \sigma_i^{\mathsf{left}}$, and the case $\sigma_i^{\mathsf{minmod}} = 0$ is trivial.

1D MUSCL-Hancock procedure: Now, let us now detail the MUSCL-Hancock method, as used in the CanoPcode. We recall that the method decomposes into three steps:

1. linear reconstruction of the solution using MUSCL method,
2. a prediction step at an intermediate time step $\Delta t/2$,
3. interfacial flux computation using the user's selected flux formula.

Linear reconstruction

We recall that Λ is the change of variables from the set *reconstructed variables* to the conservative variables: $\Lambda : \mathbf{U} \mapsto \mathbf{W}$. At time t^n and in the local cell $\mathcal{K}_i$, we compute the slope σ_i^n by using the minmod formula (6.9), and the first intermediate left and right values $\mathbf{U}^n_{i,i-\frac{1}{2}}$ and $\mathbf{U}^n_{i,i+\frac{1}{2}}$ using (6.8). We will now go back to the conservative variables and define: $\mathbf{W}^n_{i,i-\frac{1}{2}} = \Lambda\left(\mathbf{U}^n_{i,i-\frac{1}{2}}\right)$, $\mathbf{W}^n_{i,i+\frac{1}{2}} = \Lambda\left(\mathbf{U}^n_{i,i+\frac{1}{2}}\right)$.

Prediction step

At a fictitious intermediate time step $t^{n+\frac{1}{2}} = t^n + \frac{1}{2}\Delta t$, we evaluate two new predicted variables using the following formulae:

$$\begin{aligned}\mathbf{W}^{n+\frac{1}{2}}_{i,i-\frac{1}{2}} &= \mathbf{W}^n_{i,i-\frac{1}{2}} - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathrm{F}(\mathbf{W}^n_{i,i+\frac{1}{2}}) - \mathrm{F}(\mathbf{W}^n_{i,i-\frac{1}{2}})\right), \\ \mathbf{W}^{n+\frac{1}{2}}_{i,i+\frac{1}{2}} &= \mathbf{W}^n_{i,i+\frac{1}{2}} - \frac{1}{2}\frac{\Delta t}{\Delta x}\left(\mathrm{F}(\mathbf{W}^n_{i,i+\frac{1}{2}}) - \mathrm{F}(\mathbf{W}^n_{i,i-\frac{1}{2}})\right).\end{aligned}$$

We can note that in this step, one does not need to consider the values of cells $\mathcal{K}_{i-1}$ and $\mathcal{K}_{i+1}$.

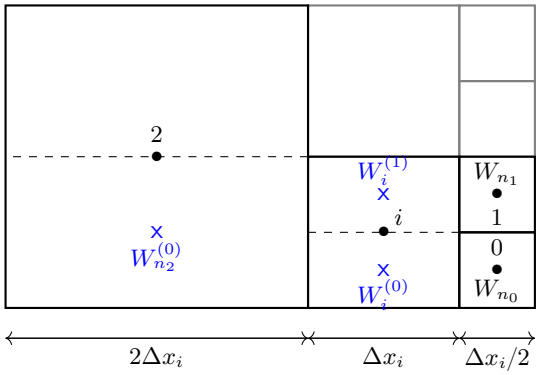
Remark 31. *In Berthon (2006), this step is considered as the resolution of two Riemann problems inside the cell $\mathcal{K}_i$: this is how the L_1-stability is proved.*

Flux computation

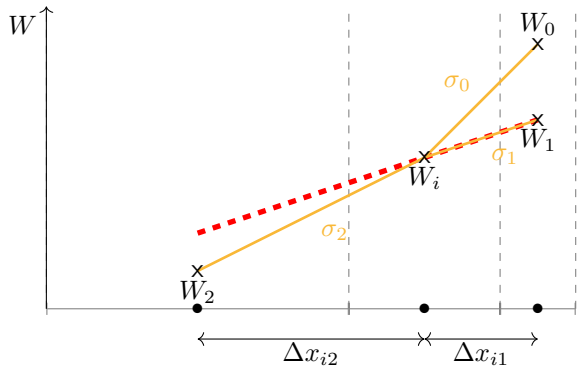
Let $\Phi(\mathbf{W}_L, \mathbf{W}_R)$ be the numerical flux, used for first-order finite volume scheme. The 1D MUSCL-Hancock scheme finally writes:

$$\mathbf{W}_i^{n+1} = \mathbf{W}_i^n + \frac{\Delta t}{\Delta x}\left(\Phi\left(\mathbf{W}^{n+\frac{1}{2}}_{i-1,i-\frac{1}{2}}, \mathbf{W}^{n+\frac{1}{2}}_{i,i-\frac{1}{2}}\right) - \Phi\left(\mathbf{W}^{n+\frac{1}{2}}_{i,i+\frac{1}{2}}, \mathbf{W}^{n+\frac{1}{2}}_{i+1,i+\frac{1}{2}}\right)\right). \tag{6.11}$$

Time step evaluation



(a) Subdivision of cells for a slope reconstruction at each face and subface.



(b) Illustration of the *minmod* slope limiter on an unstructured 1-D mesh.

Figure 6.4: *Illustration of the one-dimensional procedure for slope reconstruction.*

As mentioned previously, some care should be taken when computing Δt. Indeed, to ensure a L1-stability of the scheme, one can ensure L1-stability for each of its sub-steps. This means that we must ensure that any intermediate variable $\mathbf{W}$ satisfies $\mathbf{W} \in \Omega$. In practice, we will use the reconstructed variables $\mathbf{V}^n_{i,i-\frac{1}{2}}$ and $\mathbf{V}^n_{i,i+\frac{1}{2}}$ to locally estimate the eigenvalues λ_i^k of the Jacobian matrix and compute the time step according to the following CFL condition:

$$\forall i, \forall k, \quad \frac{|\lambda_i^k|\Delta t}{\Delta x} \leqslant \frac{1}{2}. \tag{6.12}$$

6.3.3 Application to non-conforming grids

Up to now, we have presented methods for 1D problems, where cells get a left and right neighbor. These methods can easily be extended to cartesian 2D and 3D grids, using dimensional splitting and solving a 1D problem in each direction, or simply reconstructing the slopes using the classic minmod formula. Now, let us consider cartesian non-conforming meshes, as presented at the beginning of this chapter. In these configurations, we can consider each direction individually, with the particularity that multiple sub-faces can be associated to each direction.

In the CanoPcode, we have chosen to consider a very rough approach: it consists in computing a slope for each direction (which is a mixing between the mono-slope and multi-slope approaches, as developed in Colella (1990)). Although this approach is not TVD, we will see with that the numerical results of Chapter 9 are however good.

In each direction q, let us consider all faces and sub-faces of a cell $\mathcal{K}_i$, and let us compute the slopes $(\sigma_k)_{k\in\mathcal{N}_q(\mathcal{K}_i)}$ using the values in cell $\mathcal{K}_i$ and in all its neighbors $\mathcal{N}_q(\mathcal{K}_i)$. We note that the 1D mesh considered here has not a fixed space step Δx: we must take into account the real distances separating the cells centers. In CanoP, we use a projection of this distance in the direction where the slope is computed. For instance in Figure 6.4, the space step Δx_{i1} is computed by $\Delta x_{i1} = \frac{1}{2}\left(\Delta x_i + \frac{1}{2}\Delta x_i\right) = \frac{3}{4}\Delta x_i$.

Then we apply the minmod limiter to these slopes two by two, to ensure 1D TVD. The MUSCL-Hancock procedure is thus applied in each direction separately.

We can note that this procedure is not optimal: although it shows good stability, the order of the scheme is less than 2, and degenerates easily to a first order scheme due to minmod limiter and due to the large probability that two slopes are of contrary sign within all the slopes associated with one cell. There is space for improvement of the considered method.

6.3.4 Summary of the finite volume methods for non-conforming grids

In the previous two sections, we have presented some methods to solve balance equations such as (6.1) on non-conforming grids: how fluxes are computed for each sub-face, some boundary conditions, second-order scheme.

Moreover, we take advantage of the Cartesian-like structure of the considered non-conforming grids to use some standard methods associated with the finite volume schemes, like the dimensional splitting strategies or 1D slope reconstruction.

The cartesian character of the grid is not general however. Indeed, to better comply with complex domain geometries, the cells may be deformed and splitting methods are not adapted there. This is also the case when considering triangular grids. In these cases, the unsplit strategy is used and multidimensional slope reconstructions must be designed.

6.4 Refinement strategies for AMR methods

By using AMR, we are looking for a method which is able to preserve the accuracy of a numerical solution while reducing its computational cost by *compressing* some parts of the spatial grid. This compression procedure involves the coarsening of some cells of the mesh and the refinement of other cells, according to the evolution of the numerical solution. Thus, the two following aspects must be considered:

- the choice of a criterion, according to which cells are refined or coarsened,
- the computation of the numerical solution in the newly created cells.

The choices that are made to treat these two points can greatly influence the accuracy of the numerical solution and the computational cost of the simulation. Similarly to the choice of a numerical scheme, one is looking for the best compromise between accuracy and computational cost.

Remark 32. *In this paragraph, we are not talking about the computational overhead of AMR codes in comparison with a code based on cartesian grids. This will be discussed in Chapter 7. We assume here that, whenever a mesh is compressed (i.e. coarsened out of the spatial zones of interest), the computational costs are reduced comparing with the resolution on a uniform fine grid with the same level of accuracy. This statement obviously needs to be refined since mesh handling also has a cost.*

6.4.1 Heuristic refinement criteria

The question of choosing a *refinement criterion* comes down to ensure that the quality of the solution is not damaged by coarser zones in the spatial domain, while getting as few cells as possible. In Introduction, we have mentioned multi-resolution methods (Harten (1994)), and AMR methods with criteria based on the truncation error of the numerical solution (Berger and Oliger (1984); Freret, Ivan, Sterck, and Groth (2017)). Here, we focus on an approach based on heuristic criteria, and more specifically on the gradients of some physical quantities.

The idea of the heuristic criteria based on the gradients of the numerical solution is to get a fine mesh where the solution presents strong variations and coarser cells where the solution is nearly uniform. An example of adapted grid for a variable ϕ is given in 1D in Figure 6.5 (blue grid).

In a first approach, the local discrete gradients are estimated by:

$$\boldsymbol{\nabla}_{i+\frac{1}{2}}\phi = 2\frac{\phi_{i+1}-\phi_i}{\Delta x_i + \Delta x_{i+1}}.$$

One can also consider dimensionless versions of these gradients, for non-zero values of the variables:

$$\tilde{\boldsymbol{\nabla}}_{i+\frac{1}{2}}\phi = \frac{2}{\phi_m}\frac{\phi_{i+1}-\phi_i}{\Delta x_i + \Delta x_{i+1}}, \quad \phi_m = \max\left(|\phi_i|, |\phi_{i+1}|\right) + \epsilon. \tag{6.13}$$

with $\epsilon \ll 1$ and $\epsilon > 0$.

Now, the question is to use this quantity to control the local cell size, and thus the local numerical error. A first option would be to associate to each value of the dimensionless gradient $\tilde{\boldsymbol{\nabla}}_{i+\frac{1}{2}}\phi$ a spatial step $f : \tilde{\boldsymbol{\nabla}}_{i+\frac{1}{2}}\phi \mapsto \Delta x$, where f is a decreasing step function, which is bounded from below. f will then assign the minimum cell size to cells with gradients values above a given threshold.

A second option is based only on the dimensionless difference of the numerical solution between two cells:

$$\tilde{\Delta}_{i+\frac{1}{2}}\phi = \frac{\phi_{i+1} - \phi_i}{\phi_i}. \tag{6.14}$$

The norm of this quantity is then compared to a fixed threshold ξ. If $|\tilde{\Delta}_{i+\frac{1}{2}}\phi| > \xi$, then the local cell must be refined. Else, it can be coarsened.

In CanoP, the second option is retained. Let us justify this choice by comparing the two options in two configurations encountered during a simulation, namely the grid adaptation according to the initial condition, which defines the initial grid, and the grid adaptation during the simulation.

Initial condition: the initial condition of a simulation is often provided by an analytical solution. Let us assume that the initial condition is first given on a uniform grid, with the smallest possible cells.

- With the first option, the procedure consists in visiting all cells of the grid, in computing the local gradients using (6.13), and then using the step function f to determine the appropriate size of the cell. The grid is then modified accordingly and the analytical solution is computed in each new cell.
- With the second option, all cells are also visited, the quantity (6.14) is computed, compared to the threshold ξ and possibly marked for coarsening. Then, the mesh is modified and all cells that have been marked are coarsened of one level. The solution in these cells is given by the analytical solution. This procedure must be done recursively, until coarsening is no more possible.

We can see that, in theory, the first option only necessitates one step of the coarsening procedure, while the second option is recursive and needs multiple steps. However, in practice, due to mesh management issues, and especially with *cell-based* grids, a simple way to adapt the grid according to the first option is to recursively coarsen the mesh.

Mesh adaptation during the simulation: now, the numerical solution is given at a time t^n and we would like to refine or coarsen zones where this solution presents steep gradients, or on the contrary is quite uniform.

- With the first option, all cells are visited, local gradients are computed and a corresponding cell size is prescribed by the function f. Mesh is then changed accordingly and the numerical solution is updated according to the update formulae discussed in the next paragraph.
- With the second option, all cells are visited, the quantity (6.14) is computed, compared to the threshold ξ, and cells are marked for refinement or coarsening. Mesh is then changed: cells that have been marked for refinement are refined by one level, cells that have been marked for coarsening are coarsened by one level.

The first method ensures that the grid is effectively well adapted to the numerical solution, according to the definition of the step function f. However, as for the case of the initial solution, this mesh adaptation would necessitate multiple iterations of grid adaptation. On the contrary, the second approach requires only one iteration of mesh adaptation, but the new grid may not be entirely satisfactory (if a new adaptation procedure were applied, the grid would be modified again). For practical reasons, we do not want to perform recursive mesh adaptation during the simulation. Thus, the option based on the difference between two cells (6.14) is as accurate as the the option based on the gradient of the solution (6.13). Considering that, dealing with a threshold ξ is simpler than dealing with the step function f. We finally retain the criterion (6.14) in our adaptation procedure.

In cases of systems of equations, we must determine on which variables, the refinement criterion must rely. With a heuristic approach, the choice of the variables of interest will depend on the problem to be solved. For instance in low-speed interfacial flows, where neither compressible effects nor turbulence influence the flow, one can afford a fine mesh near the interface and assume that a coarser mesh is enough

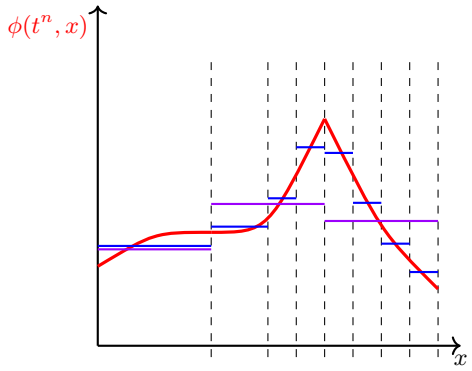


Figure 6.5: *Continuous function in red, uniform coarse mesh in purple, adapted mesh in blue.*

to solve the zones of low velocity variations. On the contrary, for highly compressible flows, where shocks and rarefactions may occur, or when one is interested in acoustic wave propagation, one will certainly have to choose pressure and velocity variations as criteria.

Multiple thresholds can be used when multiple variables are taken into account in the criterion. Equivalently, one can define a unique function $\mathbf{W} \mapsto C(\mathbf{W})$ in each cell $\mathcal{K}_i$, that will be compared with a unique threshold ξ. This function is thus defined by:

$$C(\mathbf{W})_i = \max_{j \leqslant N} \left\{ \alpha^{(j)} D(\phi^{(j)})_i \right\}, \tag{6.15}$$

where N is the number of variables in $\mathbf{W}$, α_j is a series of user-defined coefficients and:

$$D(\phi^{(j)})_i = \max \left\{ \tilde{\Delta}_{i,k} \phi^{(j)} \,|\, k \in \mathcal{N}(i) \right\}.$$

Finally, one can use two thresholds for deciding whether to refine or to coarsen, so as to avoid constant changes of the mesh that would occur when the value of the criterion $C(\mathbf{W})_i$ is close to a unique threshold. These thresholds ξ_{refine} and ξ_{coarsen} are then such that:

$$\xi_{\text{refine}} > \xi_{\text{coarsen}} \tag{6.16}$$

For each simulation presented in Chapter 9, details on the criteria and thresholds will be given.

6.4.2 Mesh adaptation procedure

Each time the grid must be adapted, one will first go through each cell $\mathcal{K}_i$ of the current mesh, and determine if it has to be refined or coarsened. Assuming a criterion and thresholds are given, the procedure is then:

1. Compute $C(\mathbf{W})_i$,
2. Compare $C(\mathbf{W})_i$ with ξ_{refine} and ξ_{coarsen}:
   - if $C(\mathbf{W})_i > \xi_{\text{refine}}$, then cell $\mathcal{K}_i$ is marked for refinement,
   - else if $C(\mathbf{W})_i < \xi_{\text{coarsen}}$, then cell $\mathcal{K}_i$ is marked for coarsening,
   - else, it is marked for no necessary change.
3. Mesh is then modified and new values inside each cell are set according to the procedures that are described in the following paragraphs.

Coarsening cells: When coarsening a set of cells into a larger one (as illustrated in 2D in Figure 6.6), the only conservative way to set the new variables values of cell $\mathcal{K}_i$ is to average the original cells values

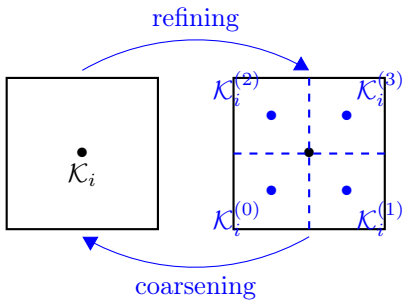


Figure 6.6: *Sketch of a 2D cell refinement or coarsening.*

weighted by the volume of these cells. For example for a variable ϕ, and a coarsening of N old cells into a new one, this writes:

$$\phi(\kappa_i) = \frac{1}{|\kappa_i|} \sum_{j<N} \phi\left(\kappa_i^{(j)}\right) \left|\kappa_i^{(j)}\right|$$

When solving systems of equations, in order to conserve quantities such as mass, momentum, energy during the coarsening process, it seems obvious to do the averaging on the conservative variables. An other option used to set the values in the coarsened cells is discussed in Appendix D, where gravity effects are taken into account.

Refining cells: There exist different strategies used when refining cells. The simplest one consists in simply copying the old variable values of cell κ_i into the newly created cells. This ensures an automatic conservation of the conservative variables, and has no impact on the numerical solution before the next time step in the numerical resolution. This is done for instance in Golay, Ersoy, Yushchenko, and Sous (2015) when first-order finite volume schemes are used.

However, the point of refining cells is precisely to improve the local accuracy of the solution. Reconstruction techniques can thus be considered, so as to set the values into each new cell according to the form of the solution in the neighborhood of these new cells. This information may be based on the estimation of the local gradients (using directional slope reconstructions or least squares methods). In Ersoy, Golay, and Yushchenko (2013), it is proposed to use the gradients computed during the second-order MUSCL scheme. Or, when gravity source terms are present, hydrostatic profiles may need to be reconstructed in the new cells. All these reconstruction techniques should be used so as to preserve the conserved quantities.

In the following, except for the gravity cases, which are discussed in Appendix D, we will take the simple first option: the solution in each new fine cell is equal to the solution in the old coarser cell.

6.4.3 Conclusion on refinement strategies

In this section, we have proposed methods for adaptive mesh refinement that answer the questions: *Where should the grid be adapted?* and *How can we do it?*:

- a heuristic criterion, based on the steepness of the numerical solution, involves to compute the quantity (6.15). Then, this quantity is compared to a threshold ξ: when larger, the local cell is refined, otherwise, it can be coarsened. The variables involved in (6.15) depend on the numerical test,
- the solution which is set in new coarse cells is obtained by the averaging of the conservative values from the set of the old finer cells. The solution which is set in a set of fine cells is a simple copy of the solution from the old coarser cell.

These choices have an influence on the compression error of the numerical solution (defined in the Introduction of Part II), but it is complex to quantify this error and to try to control it when using a

heuristic criterion as presented above. Indeed, the compression error will depend on the values given to the refinement and coarsening thresholds, as well as on the simulated configuration. Some analysis of the compression error is done in Chapter 9 on an advection test case. To deal with this issue, the multi-resolution approach (of Harten (1994) for example) is probably the method with the most developed mathematical grounds.

However, some improvement of the methods presented here can be considered for future studies. First, the second derivative of the solution could be used to better characterize the non-regularity of the solution. Indeed, with the finite volume schemes, and more particularly when second-order reconstructions are used, the first-order variations of the solution are already well resolved. Moreover, the second-order variations of the solution is also important when solving equations with diffusion terms (see for example Anthonissen, Bennett, and Smooke (2005) and references to their previous works therein).

Second, in Golay, Ersoy, Yushchenko, and Sous (2015), a criterion based on the production of entropy is used to characterize the spatial zones where refinement is needed. Such criteria could be applied to our systems of equations, for which the entropy function is known.

The original AMR method of Berger and Oliger (1984) was based the truncation error of the numerical solution. This error was estimated by computing the solution over two different levels of refinement, and comparing the two results. Similar ideas are found in Freret, Ivan, Sterck, and Groth (2017).

6.5 Low-Mach flows and Godunov-type schemes

The last numerical aspect we focus on in this chapter concerns the simulation of 2D and 3D fluid flows, in a low-Mach number regime, i.e. nearly mechanically incompressible flows.

In this context, one characteristic quantity is the Mach number M, defined by the ratio of the fluid velocity over the speed of sound:

$$M = \frac{|\mathbf{u}|}{c}. \tag{6.17}$$

For instance, with $M < 1$, flows are *subsonic*, for $M \approx 1$ they are transonic, for $M > 1$ they are supersonic. One also encounters the classification of weakly compressible or weakly dilatable flows for $M < 0.1$, and mechanically incompressible for $M < 0.01$. We will recall in the following paragraph why flows are considered as nearly incompressible when the Mach number is very low.

It is furthermore well-known that classic Godunov-type schemes for compressible equations (Euler or Navier-Stokes) give bad results in the low-Mach regime (Guillard and Viozat (1999); Clerc (2000); Paillere, Viozat, Kumbaro, and Toumi (2000)). The origin of this defect is not yet fully understood: some analysis for specific initial and boundary conditions and for the barotropic Euler equations show that finite volume schemes develop a large numerical viscosity, proportional to M, that leads to a poor spatial numerical resolution.

Different methods are thus being developed to get rid of this defect and to enable simulations of all-Mach regime flows. For the barotropic two-fluid equations we are considering in this thesis, we will present two of the possible approaches that have helped to improve the numerical results for interfacial flows. Some artifacts in the numerical solutions still have to be solved, as we will see in Chapter 9.

6.5.1 Theoretical analysis of the problem

Let us first give some ideas on the continuous equations, and their asymptotic limit when the Mach number tends to zero. The rigorous derivation of the incompressible equations from the compressible ones and the convergence of the solutions from the first system towards the second one in the low-Mach number limit is still a research topic, since only specific configurations (among them the barotropic equations, periodic or wall boundary conditions...) are now well-known. We refer to the various works of Klainerman and Majda (1981); Schochet (1986); Schochet (1994); Lions and Masmoudi (1998); Secchi (2000); Genieys and Massot (2001) for deeper studies on these aspects.

From now, let us consider a general form of barotropic fluid equations written as:

$$\left\{\begin{array}{rcrcl} \partial_t(\rho\mathbf{u}) & + & \mathrm{div}_x\left(\rho\mathbf{u}\mathbf{u}^T + p\,\mathbf{Id}\right) & = & \mathbf{0} \\ \partial_t\rho & + & \mathrm{div}_x(\rho\mathbf{u}) & = & 0 \\ \partial_t\mathbf{V} & + & \mathbf{u}\cdot\boldsymbol{\nabla}_x\mathbf{V} & = & \mathbf{0} \end{array}\right. \tag{6.18}$$

and the pressure p is a function of ρ and $\mathbf{V}$: $p(\rho,\mathbf{V})$, with $\partial_\rho p = c^2(\rho,\mathbf{V})$ the definition of the isentropic speed of sound and $\partial_{\mathbf{V}} p = \beta(\rho,\mathbf{V})$.

Dimensionless equations

Let us consider smooth solutions to system (6.18), and write:

$$\rho = R\tilde{\rho}\,,\quad \mathbf{u} = U\tilde{\mathbf{u}}\,,\quad p = P\tilde{p}.$$

Moreover, as we are interested in slow flows, let us consider large time and space scales defined by:

$$t = T\tilde{t}\,,\quad x = L\tilde{x}\,,\quad \text{with}\quad U = L/T.$$

For the sake of simplicity, we assume that the variables $\mathbf{V}$ are uniform and constant. Then the problem comes down to study the barotropic Euler equations, with the relation:

$$P = C^2(R,\mathbf{V})R$$

where C is a characteristic speed of sound.

Remark 33. *In two-phase flows, and particularly for interfacial flows, the variables ρ and $\mathbf{V}$ are spatially discontinuous. Getting dimensionless equations seems then more complex (a two-phase low-Mach number system is proposed in Dellacherie (2005) by considering two separate zones of the flow made of pure phases). In some cases, such as disperse phase flows, smooth profiles of $\mathbf{V}$ can however be encountered. One will assume here that whatever the value of $\mathbf{V}$, the Mach-number regime is the same ($M < 0.01$ for instance).*

Using the previous definitions, the dimensionless system of equations can easily be deduced from (6.18) and writes:

$$\left\{\begin{array}{rcrcl} \partial_{\tilde{t}}(\tilde{\rho}\tilde{\mathbf{u}}) & + & \mathrm{div}_{\tilde{x}}\left(\tilde{\rho}\tilde{\mathbf{u}}\tilde{\mathbf{u}}^T\right) + \dfrac{1}{M^2}\boldsymbol{\nabla}_{\tilde{x}}(\tilde{p}) & = & \mathbf{0} \\ \partial_{\tilde{t}}\tilde{\rho} & + & \mathrm{div}_{\tilde{x}}(\tilde{\rho}\tilde{\mathbf{u}}) & = & 0 \\ \partial_{\tilde{t}}\mathbf{V} & + & \tilde{\mathbf{u}}\cdot\boldsymbol{\nabla}_{\tilde{x}}\mathbf{V} & = & \mathbf{0} \end{array}\right. \tag{6.19}$$

with $M = U/C$ the characteristic Mach number. The low-Mach regime is defined by the two following conditions:

- $M \ll 1$,
- the variations of $\boldsymbol{\nabla}_{\tilde{x}}\tilde{p}$ remain of the order of M^2.

Incompressible equations

From system (6.19) and by letting M tend to zero, it is then possible to derive the incompressible equations for a flow with periodic or wall boundary conditions.

As in Dellacherie (2010), let us assume that the dimensionless pressure and velocity write:

$$\tilde{p}(\tilde{t},\tilde{x}) = \tilde{p}^{(0)}(\tilde{t},\tilde{x}) + M\tilde{p}^{(1)}(\tilde{t},\tilde{x}) + M^2\tilde{p}^{(2)}(\tilde{t},\tilde{x}) + O(M^3),$$
$$\tilde{\mathbf{u}}(\tilde{t},\tilde{x}) = \tilde{\mathbf{u}}^{(0)}(\tilde{t},\tilde{x}) + M\tilde{\mathbf{u}}^{(1)}(\tilde{t},\tilde{x}) + M^2\tilde{\mathbf{u}}^{(2)}(\tilde{t},\tilde{x}) + O(M^3).$$

From the momentum equation of system (6.19), we see that the spatial derivatives of $\tilde{p}$ are null for the zeroth and first orders in $M \ll 1$. We then write:

$$\tilde{p}(\tilde{t},\tilde{x}) = \tilde{p}^{(0)}(\tilde{t}) + M^2\tilde{p}^{(2)}(\tilde{t},\tilde{x}) + O(M^3)$$

and due to the barotropic state laws and the uniformity of $\mathbf{V}$, we can assume that:

$$\tilde{\rho}(\tilde{t},\tilde{x}) = \tilde{\rho}^{(0)}(\tilde{t}) + M^2\tilde{\rho}^{(2)}(\tilde{t},\tilde{x}) + O(M^3)$$

Now, the mass equation can be written at zeroth order in M:

$$d_{\tilde{t}}\tilde{\rho}^{(0)} + \tilde{\rho}^{(0)}\mathrm{div}_{\tilde{x}}\tilde{\mathbf{u}}^{(0)} = 0,$$

or

$$\mathrm{div}_{\tilde{x}}\tilde{\mathbf{u}}^{(0)} = -\frac{d_{\tilde{t}}\tilde{\rho}^{(0)}}{\tilde{\rho}^{(0)}}. \tag{6.20}$$

When integrating this equation on the whole spatial domain Ω, the left-hand side term becomes:

$$\begin{aligned}\int_\Omega \mathrm{div}_{\tilde{x}}\tilde{\mathbf{u}}^{(0)} &= \int_{\partial\Omega}\tilde{\mathbf{u}}^{(0)T}\mathbf{n}\\ &= 0\end{aligned}$$

due to the boundary conditions (wall boundary or periodic boundary conditions).

The right-hand side can be written as:

$$\int_\Omega \frac{d_{\tilde{t}}\tilde{\rho}^{(0)}}{\tilde{\rho}^{(0)}} = \frac{d_{\tilde{t}}\tilde{\rho}^{(0)}}{\tilde{\rho}^{(0)}}$$

and assuming that $\tilde{\rho}^{(0)} > 0$, one finally finds that:

$$d_{\tilde{t}}\tilde{\rho}^{(0)} = 0.$$

Density and pressure finally write:

$$\begin{aligned}\tilde{p}(\tilde{t},\tilde{x}) &= \tilde{p}^{(0)} + M^2\tilde{p}^{(2)}(\tilde{t},\tilde{x}) + O(M^3)\\ \tilde{\rho}(\tilde{t},\tilde{x}) &= \tilde{\rho}^{(0)} + M^2\tilde{\rho}^{(2)}(\tilde{t},\tilde{x}) + O(M^3)\end{aligned}$$

where $\tilde{p}^{(0)}$ and $\tilde{\rho}^{(0)}$ are constant quantities. Coming back to equation (6.20), one can see that at zeroth order in M, the velocity field satisfies:

$$\mathrm{div}_{\tilde{x}}\tilde{\mathbf{u}}^{(0)} = 0.$$

Finally, when $M \to 0$, the first order in M system writes:

$$\left\{\begin{array}{lcccc} \partial_{\tilde{t}}\left(\tilde{\mathbf{u}}^{(0)}\right) & + & \mathrm{div}_{\tilde{x}}\left(\tilde{\mathbf{u}}^{(0)}\tilde{\mathbf{u}}^{(0)T}\right) + \dfrac{1}{\tilde{\rho}^{(0)}}\boldsymbol{\nabla}_{\tilde{x}}\left(\tilde{p}^{(2)}\right) & = & \mathbf{0}\\ & & \mathrm{div}_{\tilde{x}}\left(\tilde{\mathbf{u}}^{(0)}\right) & = & 0\\ \partial_{\tilde{t}}\mathbf{V} & + & \tilde{\mathbf{u}}^{(0)}\cdot\boldsymbol{\nabla}_{\tilde{x}}\mathbf{V} & = & \mathbf{0}\end{array}\right. \tag{6.21}$$

This formulation is close to an (inviscid) incompressible system of equations. Now, the pressure variable is not related to the thermodynamics of the system anymore. When solving the incompressible equations, one will have to solve both $\boldsymbol{\nabla}_{\tilde{x}}\left(\tilde{p}^{(2)}\right)$ and $\tilde{\mathbf{u}}^{(0)}$.

Solving the incompressible equations is a method largely used in numerous flows configuration. We will describe later the numerical methods that are associated with this approach. Now, to better understand why the compressible equations and their finite volume schemes are so difficult to use in the low-Mach configurations, we propose to use the Hodge-Helmholtz theory, as in Dellacherie (2010).

Hodge theory

The Hodge, or Hodge-Helmholtz decomposition is, for a given vector field, its unique decomposition into an irrotational part, a solenoidal part and, depending on the domain topology and boundary conditions, a harmonic part (Hodge (1989); Lemoine (2014)). Let Ω be an open subspace of $\mathbb{R}^n$ and $\mathbf{v}$ a vector field on Ω. Then, the Hodge decomposition of $\mathbf{v}$ is:

$$\mathbf{v} = \mathbf{v}_\theta + \mathbf{v}_\psi + \mathbf{v}_h$$

such that $\mathbf{rot}\,(\mathbf{v}_\theta) = \mathbf{0}$ and $\mathrm{div}(\mathbf{v}_\psi) = 0$. The last component is not uniquely defined. For examples of $\mathbf{v}_h$, the reader is referred to Lemoine (2014).

The subspaces of irrotational and solenoidal vector fields are respectively noted $\mathcal{E}_\theta, \mathcal{E}_\psi \subset \left(L^2(\Omega)\right)^n$. In the following, we will assume that there is no harmonic component. One can then show that $\mathcal{E}_\theta$ and $\mathcal{E}_\psi$ are orthogonal ($\mathcal{E}_\theta \perp \mathcal{E}_\psi$) and their sum is the entire space of vector fields:

$$\mathcal{E}_\theta \oplus \mathcal{E}_\psi = L^2(\Omega).$$

For the incompressible equations, the vector field $\mathbf{v}$ corresponds to the velocity field, and the incompressible solution remains in the solenoidal subspace $\mathcal{E}_\psi$.

In Dellacherie (2010), the low-Mach formulation for the Euler equations (6.19) is written in the form:

$$\partial_t q + \mathcal{H} q + \frac{\mathcal{L}}{M} q = 0 \tag{6.22}$$

where $\mathcal{H}$ and $\mathcal{L}$ respectively stand for the convective and acoustic operators, and the variable q is made of first order variations of ρ and the velocity field $\tilde{\mathbf{u}}^{(0)}$: $q = \left(r, \tilde{\mathbf{u}}^{(0)}\right)^T$. On a torus (a domain with periodic boundary conditions everywhere), the solenoidal subspace $\mathcal{E}_\psi$ is associated with the subspace $\mathcal{E}$ generated by the solutions q from the kernel of the acoustic operator $\mathcal{L}$ (i.e. such that $\mathcal{L}q = 0$). Indeed, these solutions verify the divergence-free condition for the velocity field, $\mathrm{div}_{\tilde{x}} \tilde{\mathbf{u}}^{(0)} = 0$, and a uniform profile for r: $\boldsymbol{\nabla}_{\tilde{x}} r = \mathbf{0}$. The subspace $\mathcal{E}$ is called the *well-prepared* subspace, and its orthogonal complement for the Hodge decomposition is noted $\mathcal{E}^\perp$.

Any solution q of (6.22) will then be decomposed as:

$$q = \hat{q} + q^\perp, \quad \text{with} \quad \hat{q} \in \mathcal{E} \quad \text{and} \quad q^\perp \in \mathcal{E}^\perp$$

According to Dellacherie (2010), and from a result of Schochet (1994), if any initial solution $q(t = 0, x)$ is *well-prepared*, in the sense that

$$\|q - \hat{q}\| = O(M),$$

with $\|\cdot\|$ the norm for the Hilbert space $\left(L^2(\Omega)\right)^d$ associated with the scalar product $q_1, q_2 \mapsto \int_\Omega q_1^T q_2$, then at time $t > 0$, solution $q(t, x)$ of (6.19) is close to the solution $\tilde{q}$ of the incompressible equations by:

$$\|q - \tilde{q}\| = O(M).$$

Study of cartesian grids and divergence operators

Up to the end of the previous paragraph, we have focused on the continuous equations, and their low-Mach number limit. From a discrete point of view, we expect the solutions to satisfy the same kind of properties: the numerical compressible solution should stay close to the incompressible solution in the low-Mach number regime.

An analysis of a Godunov-like scheme for the compressible dimensionless system (6.22) reveals that the equivalent continuous equations (see definition in Chapter 4) write:

$$\partial_t q + \mathcal{H} q + \frac{\mathcal{L}}{M} q = \mathcal{B}_\nu q \tag{6.23}$$

where $\mathcal{B}_\nu$ is a non-isotropic diffusive operator, associated with a numerical viscosity ν, whose magnitude is of order $O(C\frac{\Delta x}{M})$ where Δx is the spatial discrete step. Probably due to this operator, an initially well-prepared solution is no longer close to the incompressible solution for times $t > 0$. This was proved for the linear case in Dellacherie (2010). On the other hand, cures to Godunov schemes based on the reduction of this numerical viscosity seem to improve the numerical results, even for the non-linear equations.

Remark 34. *It is possible to define a discrete well-prepared subspace $\mathcal{E}^{\Delta x}$ for a Godunov-like scheme. It is then the kernel of the operator $\frac{\mathcal{L}}{M} - \mathcal{B}_\nu$. For a cartesian discretization, namely a grid made of quadrangular cells, $\mathcal{E}^{\Delta x}$ is defined as the set of solutions q such that $\partial_{x_k} \tilde{u}_k^{(0)} = 0$, where k are indices for the spatial directions. This subspace is very small. It is included into the continuous well-prepared subspace $\mathcal{E}$, but is far from being equal to it:*

$$\mathcal{E}^{\Delta x} \subsetneq \mathcal{E}.$$

In the case of triangular meshes, however, $\mathcal{E}^{\Delta x}$ is different: on these grids, nearly incompressible flows can be computed using the classic forms of Godunov schemes and compressible Euler equations. This seems not true however when considering thermal effects, as can be observed in the numerical results of Dellacherie (2010).

6.5.2 A review of existing cures

Before presenting the two solutions that were retained and implemented in CanoP, we present here some methods usually employed for computing low-Mach-number flows.

Incompressible equations As we have seen previously, the incompressible systems of equations can be considered as the asymptotic limit of the compressible ones when $M \to 0$. Incompressible equations are widely used for simulating hydrodynamics flows in the low-Mach regime. However, by their definition, most of the standard methods can not be applied in a flow where different Mach-regime must be taken into account. Methods based on weakly compressible flows (the incompressible equations are slightly modified so as to account for density variations) have been developed in Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) and are compared with the solution of compressible equations in Chapter 10.

Godunov schemes on staggered grids One of the most widespread method for dealing with low-Mach flows and compressible equations consists in considering that velocity and pressure are discretized at different locations: while the first one is given at the cells interfaces, the second one is taken at the centers of the cells. This approach is inspired from the resolution of the incompressible equations, where staggered grids are necessary for sakes of stability (Hirsch (1990)). With compressible equations, it was used in Woodward and Colella (1984); Liou (1996); Li and Gu (2008); Bernard-Champmartin and De Vuyst (2014) and with a different naming, in Grenier, Vila, and Villedieu (2013) (where the centered discretization of the pressure term comes down to a staggered grid). Using a centered discretization for pressure permits to cancel out the awkward viscosity in the numerical diffusion tensor. The main drawback of this method is its need for artificial viscosity so as to be stable. We will see later that the all-regime correction comes down to a centered discretization of pressure, which is close to the staggered grids method.

Preconditioning The idea of the preconditioning method is to artificially change the compressibility of the fluid system when the Mach number tends to zero. This is done through the multiplication by a **preconditioning** matrix with the temporal operators, enabling to solve the equations at the time scale of convection and no longer at that of acoustics. This method was introduced by Turkel (1987) and was then widely developed Turkel (1993); Guillard and Viozat (1999). However, this approach is more complex to use in practice than other ones: they are not well adapted to complex boundary conditions and encounters instability issues (Martelot (2013)).

Modification of the state laws An other approach which is also based on the modification of the fluids compressibility consists in changing the thermodynamic state laws by artificially lowering the speeds of sound. This is done for instance in Chanteperdrix, Villedieu, and Vila (2002). This modification has two advantages: first, with reduced sound velocities, the Mach number is increased, and the numerical viscosity, of order $O\left(\frac{\Delta x}{M}\right)$ is thus smaller. Secondly, in explicit schemes, the time step, restricted by the CFL condition, can be larger, as the acoustic constraint is reduced. However, as mentioned in Grenier, Vila, and Villedieu (2013), changing the compressibility may not be physically relevant with respect to other effects, such as surface tension, acoustic waves... This method is then limited to some particular flow configurations.

Modified flux function Based on an analysis of the consistency errors for the Godunov-like schemes, it is possible to change the flux formulae so as to cancel the bad numerical viscosity. This method is presented in Girardin (2014); Chalons, Girardin, and Kokh (2016) and we will give a larger presentation of it below.

Triangular grids As mentioned previously, triangular or unstructured grids show good results whatever the Mach-number regime, but only with certain boundary conditions, and not with all systems of equations. This was noticed in Dellacherie (2010); Chalons, Girardin, and Kokh (2016); Grenier, Vila, and Villedieu (2013); Tsoutsanis, Kokkinakis, Könözsy, Drikakis, Williams, and Youngs (2015) and a bit further analyzed in Dellacherie, Omnes, and Rieper (2010).

From all these presented methods, we are interested more particularly in two of them:

- the triangular grid method, since it does not need any scheme modification, and in the context of CanoPcode, its use is simple (see Chapter 8),
- the modified-flux approach of Chalons, Girardin, and Kokh (2016) is also simple to implement considering our current approximate Riemann solvers (Chapter 4).

6.5.3 The all-regime scheme

The method presented in this part is issued from Chalons, Girardin, and Kokh (2016). It was developed with the objective of dealing with flows, in which the range of values for the Mach number can be large and encompass multiple flow regimes. In Chalons, Girardin, and Kokh (2016) the method was applied to the Euler equations, and in Girardin (2014), an application to a two-fluid homogeneous system is proposed.

After the presentation of the original all-regime scheme and of its effect on the reduction of the truncation error, we will propose an adaptation to the numerical methods that were presented in Chapter 4.

Lagrange-Projection scheme

The original all-regime scheme has been developed for the Lagrange-Projection scheme. This scheme is briefly presented below. Let us first write system (6.18) in the following form:

$$\left\{\begin{array}{rcrcrcl} \rho\partial_t\mathbf{u} & + & \rho\left(\mathbf{u}^T\boldsymbol{\nabla}_x\right)\mathbf{u} & + & \boldsymbol{\nabla}_x p & = & \mathbf{0} \\ \rho\partial_t\tau & + & \rho\mathbf{u}^T\boldsymbol{\nabla}_x\tau & - & \mathrm{div}_x\mathbf{u} & = & 0 \\ \rho\partial_t\mathbf{V} & + & \rho\mathbf{u}^T\boldsymbol{\nabla}_x\mathbf{V} & & & = & \mathbf{0} \end{array}\right. \tag{6.24}$$

with $\tau = \frac{1}{\rho}$ the specific volume. The blue terms correspond to the transport part of the equations, while the purple terms describe the acoustic part. The Lagrange-Projection method is in fact based on the operator splitting between the convective and the acoustic effects. It comes down to solve successively the two following subsystems:

$$\left\{\begin{array}{rcrcl} \partial_t\mathbf{u} & + & \tau\boldsymbol{\nabla}_x p & = & \mathbf{0} \\ \partial_t\tau & - & \tau\mathrm{div}_x\mathbf{u} & = & 0 \\ \partial_t\mathbf{V} & & & = & \mathbf{0} \end{array}\right. \tag{6.25}$$

which is called **Lagrangian** part, and

$$\begin{cases} \partial_t \mathbf{u} & + & \left(\mathbf{u}^T \boldsymbol{\nabla}_x\right) \mathbf{u} & = & \mathbf{0} \\ \partial_t \tau & + & \mathbf{u}^T \boldsymbol{\nabla}_x \tau & = & 0 \\ \partial_t \mathbf{V} & + & \mathbf{u}^T \boldsymbol{\nabla}_x \mathbf{V} & = & \mathbf{0} \end{cases} \tag{6.26}$$

which is the **projection** part. Introducing a mass variable as $m = \rho x$ enables to write system (6.25) in a conservative form.

In Chalons, Girardin, and Kokh (2016), the Lagrangian part (6.25) is numerically solved using Suliciu's relaxation method, and the projection part (6.26) with an upwind Finite Volume scheme.

We can note that the operator decomposition of the Lagrange-Projection method is similar to the operator decomposition in the low-Mach limit of Dellacherie (2010). It is indeed shown in Chalons, Girardin, and Kokh (2016) that the issues in the low-Mach regime come from the resolution of the acoustic or Lagrangian part of the system, while the transport part has no harmful effect.

Truncation errors for the acoustic part

The analysis of the properties of the numerical scheme is performed in Chalons, Girardin, and Kokh (2016) by first rescaling the discrete equations, as has been done for the continuous system in the previous paragraphs. This reveals the influence of the Mach number M in the flux formulae. In a low-Mach regime, where the pressure profile is smooth and of order M^2, it is then shown that the equivalent continuous equations for system (6.25), after being solved with a Suliciu's relaxation scheme over a rescaled time step $\Delta\tilde{t}$ and on a spatial discretization of space step $\Delta\tilde{x}$, writes in 1D

$$\begin{cases} \tilde{\rho}\partial_{\tilde{t}}\tilde{u} & + & \dfrac{1}{M^2}\partial_{\tilde{x}}\tilde{p} & = & O(\Delta\tilde{t}) + O\left(\dfrac{\Delta\tilde{x}}{M}\right) + O(\Delta\tilde{x}^2) \\ \partial_{\tilde{t}}\tilde{\tau} & - & \tilde{\tau}\partial_{\tilde{x}}\tilde{u} & = & O(\Delta\tilde{t}) + O(M\Delta\tilde{x}) + O(\Delta\tilde{x}^2) \\ \partial_{\tilde{t}}\tilde{\mathbf{V}} & & & = & \mathbf{0} \end{cases} \tag{6.27}$$

We can clearly see that in the momentum equation, the truncation error of order $\dfrac{\Delta\tilde{x}}{M}$ will have a nasty effect when $M \ll 1$ and especially if $M \ll \Delta x$. Something should then be done to limit the effect of this term, apart from arbitrarily decreasing the space step Δx.

Remark 35. *This 1D analysis makes appear a viscous dissipation of order $\Delta\tilde{x}/M$, such as the numerical viscosity estimated in Dellacherie (2010). While for 1D simulations, this numerical viscosity does not damage the solution, its presence in the non-isotropic dissipation tensor for 2D and 3D cases is really detrimental.*

Modification of the flux function

The analysis of the Suliciu's relaxation scheme for the acoustic system (6.25), shows that the apparition of the damaging truncation error comes from a bad estimation of the intermediate value for pressure in the Riemann solution.

Indeed, for system (6.25), the update formula for velocity writes:

$$u_i^{LAG} = u_i^n - \frac{\Delta t}{\Delta m_i^n}\left(p^*_{i+\frac{1}{2}} - p^*_{i-\frac{1}{2}}\right)$$

with the intermediate pressure $p^*_{i\pm\frac{1}{2}}$ given by:

$$p^*_{i\pm\frac{1}{2}} = \frac{1}{2}\left(p_i^n + p_{i\pm1}^n\right) \pm \frac{a}{2}\left(u_i^n - u_{i\pm1}^n\right)$$

where a is a constant (see Chapter 4). Rescaling the two previous equations gives:

$$\tilde{u}_i^{LAG} = \tilde{u}_i^n - \frac{\Delta\tilde{t}}{\tilde{\rho}_i^n \Delta\tilde{x} M^2}\left(\tilde{p}^*_{i+\frac{1}{2}} - \tilde{p}^*_{i-\frac{1}{2}}\right)$$

and

$$\tilde{p}^*_{i\pm\frac{1}{2}} = \frac{1}{2}\left(\tilde{p}^n_i + \tilde{p}^n_{i\pm1}\right) \pm \frac{\tilde{a}M}{2}\left(\tilde{u}^n_i - \tilde{u}^n_{i\pm1}\right).$$

From the last equation, we can show that the first order term in M should in fact be at second order, so as to compensate the truncation errors. The idea of the correction proposed in Chalons, Girardin, and Kokh (2016) is to change the velocity flux function, using a parameter θ in the following way:

$$p^*_{i\pm\frac{1}{2}} = \frac{1}{2}\left(p^n_i + p^n_{i\pm1}\right) \pm \frac{a\theta}{2}\left(u^n_i - u^n_{i\pm1}\right). \tag{6.28}$$

The resulting dimensionless equivalent equation for momentum then writes:

$$\tilde{\rho}\partial_{\tilde{t}}\tilde{u} + \frac{1}{M^2}\partial_{\tilde{x}}\tilde{p} = O(\Delta\tilde{t}) + O\left(\frac{\theta\Delta\tilde{x}}{M}\right) + O(\Delta\tilde{x}^2)$$

and a good tuning of θ allows to limit the harmful effects of the truncation errors.

For the Euler equations, the modified Lagrange-Projection scheme has the following properties (see details in Chalons, Girardin, and Kokh (2016)):

- it is conservative with respect to the conservative variables (density, momentum, total energy),
- under a certain condition (given in Chalons, Girardin, and Kokh (2016)), it satisfies a discrete entropy inequality,
- density remains positive and if the discrete entropy inequality is satisfied, so does the specific internal energy,
- with the choice $\theta = O(M)$, the truncation error in the low-Mach regime is of order $O(\Delta x)$.

Remark 36. *When $\theta \ll 1$, the correction in the momentum flux can be seen as a centered discretization of the pressure. This property reminds us of the approach of the staggered grids schemes.*

Application to Eulerian schemes and limits of the scheme:

The correction to the Lagrange-Projection scheme as presented above can in fact also be applied for Eulerian schemes. It simply comes down to modifying the pressure value in the solution of the local Riemann problems. For the three schemes presented in Chapter 4, this writes:

- Relax-S scheme:

$$p^*_{i\pm\frac{1}{2}} = \frac{1}{2}\left(p_i + p_{i\pm1}\right) \quad \pm \quad \theta\,\frac{a}{2}\left(u_i - u_{i\pm1}\right)$$

- Relax-B scheme:

$$p^*_{i\pm\frac{1}{2}} = \frac{a_i p_{i\pm1} + a_{i\pm1}p_i}{a_i + a_{i\pm1}} \quad \pm \quad \theta\, a_i a_{i\pm1}\frac{u_i - u_{i\pm1}}{a_i + a_{i\pm1}}$$

- HLLC flux:

$$p^*_{i\pm\frac{1}{2}} = \frac{j_i p_{i\pm1} - j_{i\pm1}p_i}{j_i - j_{i\pm1}} \quad \pm \quad \theta\, j_i j_{i\pm1}\frac{u_i - u_{i\pm1}}{j_i - j_{i\pm1}}$$

Two defects of the all-regime schemes have however been noticed:

- the first is the instability of the schemes in a 2D or 3D configuration when using a dimensional splitting, while no difficulty is encountered with an *unsplit* method. The grounds of this problem are currently under study, and a solution for the Lagrange-Projection scheme is being tested.
- the second problem is now a known numerical issue: for several low-Mach or all-regime schemes, **checkerboard modes** may appear during a simulation (see illustrations in Chapter 9). Some analysis of this issue can be found in Dellacherie (2009). In the configurations we are interested in, namely for the two-fluid models with gravity source term, these checkerboard modes appear when using a locally refined mesh, while they are not visible on uniform grids. These observations will be discussed in Chapter 9).

6.6 Conclusion

In this chapter, we have considered different numerical methods, associated with the finite volume schemes and which are standard in simulations on cartesian grids (Leveque (2004)): these are dimensional splittings, second-order reconstruction... These methods have been adapted to cartesian non-conforming grids, since such grids are used in a cell-based AMR framework.

Moreover, the AMR strategy is made of heuristic refinement/coarsening criteria, based on the first-order spatial variations of the solution, and of simple projection and averaging procedures to set the values of the numerical solution in newly adapted grids. These choices have been made for their simplicity and can be further improved. In particular, strategies that would permit to better control the compression error could be considered.

Finally, we have addressed the issue of the numerical solutions of compressible equations in the low-Mach number limit. In the AMR framework and when solving two-phase flows, this issue is critical (results are presented and compared in Chapter 9 and Chapter 10). After a rough presentation of the issue and the existing cures, we have recalled the all-regime scheme of Chalons, Girardin, and Kokh (2016), which is compatible with our equations and the finite volume strategies presented in the previous sections.

Different numerical tests involving all these methods are presented in Chapter 9, and aim at verifying their practical feasibility, robustness and accuracy.

Part III

Adaptive Mesh Refinement methods and codes

Introduction

In Berger and Collela (1989), two visions of Adaptive Mesh Refinement (AMR) are mentioned. According to the first one, the goal of AMR would be to increase the accuracy of a computation for a fixed computational cost. According to the second vision, the point of AMR would be to save computational cost while preserving the accuracy of the numerical solution. These visions could in fact apply at different levels of a numerical strategy: for example, when considering the model to solve, when choosing high-order schemes, etc. That is why adaptive resolution techniques may take several forms. Moreover, these techniques can be combined: one can adapt the space discretization, the time discretization, the order of accuracy of time/space discretization, the scales that are resolved and the scales that are modeled. All of that depend on the specific needs of the computation and the available computational resources.

More specifically, adapting the spatial discretization is a widespread technique in CFD with the use of unstructured grids, which are designed before the start of the simulation and are kept fixed during the computation. Highest resolution cells are then located in zones of complex geometry, or in zones where the flow should develop strong fluctuations of its field or sharp fronts, like near the flame front in combustion simulations. Fixed unstructured grids are interesting when the locations of these types of zones can be determined a priori and do not travel too much, and some codes, such as Yales2 code (Moureau and Lartigue (2015)) show very good computation performances.

For non-stationary or complex flows, however, zones of highest interest can move in the domain, appear and disappear through the computation, and in general cases, their location is hardly predictable. For instance, with compressible equations for flow dynamics, shocks may develop due to compression effects and then propagate. Using a fixed grid for these cases would require a high resolution in most of the spatial domain, although at a given time, the sharp front can be restricted to a localized spatial zone. Thus, one may think of using grids that adapt automatically and according to the evolution of the flow, so as to keep a high level of resolution in the zones of interest, and decrease this level elsewhere.

But handling dynamically adapting grids is more complex than fixed ones and may have inhibited their use, up to the recent development of even more efficient and scalable AMR algorithms, as well as increasing communities of users of some widespread public AMR frameworks (see a review of such block-based AMR codes in Dubey, Almgren, Bell, Berzins, Brandt, Bryan, Colella, Graves, Lijewski, Löffler, et al. (2014)).

Generalities about structured AMR

In the sequel, we will concentrate on structured AMR (SAMR) approaches, which are characterized by a resolution of the solution over rectangular grids. In this context, it seems natural to compare theses methods with fixed Cartesian grids, since they involve the same numerical approaches (finite volume or finite element schemes).

Main issues:

One of the main issues of AMR codes lies in the overheads (the excess time spent in AMR specific use in comparison with an equivalent fixed grid use) that are inherently related to each method, by comparison with a pure cartesian mesh. These overheads come from the management of the mesh

connectivities. Indeed, in a fixed cartesian mesh, cells are generally indexed using their spatial location by an (i, j, k) indexing system. Then, updating the solution in each cell of the grid is made in an efficient and even optimized manner (by traversing all indices) according to the programming language and some compilation options, the use of accelerators. In adapting and non-cartesian meshes, visiting every cell, finding the neighborhood of a cell, or partitioning the mesh are no longer as simple as operating the change of indices $i \pm 1$, $j \pm 1$ or $k \pm 1$, and may necessitate some cleverer memory management. One side-effect of this first issue is that using high-order numerical schemes, which necessitate large stencils (meaning many layers of neighboring cells), may not be obvious on AMR grids, especially in a distributed computation framework.

Another issue is related to parallel computations with AMR methods and a balanced repartition of work load between the computing units. Indeed, as the mesh is changing, the work load may also change, due to the creation of zones of finer cells or the gathering or suppression of such cells. Consequently, keeping an initial domain decomposition through the whole simulation may generate a strong load imbalance. Re-repartition techniques must then be considered. As described in Gunney and Anderson (2016), a good partitioning must:

- balance the work load between the processes,
- preserve locality of data, meaning that neighboring cells should lie either in the same subdomain or in neighboring subdomains,
- avoid communications due to large subdomain interfaces,
- be a fast procedure, since it is called regularly during the computation.

Finally, for certain types of AMR, mapping domains with complex geometry may lead to complicated strategies, like Embedded boundaries as in Schwartz, Barad, Colella, and Ligocki (2006) or mapped grids as in Brown, Chesshire, Henshaw, and Quinlan (1997); Henshaw and Schwendeman (2003). In many applicative AMR codes, however, only unit square or unit cube domains are considered (for example in Ramses code of Teyssier (2002)).

Some definitions:

Some vocabulary has developed with the AMR techniques. Let us define here some notions that will be used in the following chapters:

- **Cell:** a spatial discretization element on which the numerical solution is computed,
- **Block:** an rectangular set of cells of same size. Patch-based methods are also **block-based**. However, tree-based methods can be qualified as block-based only if the leaves of the tree are made of a block of cells.
- **Level of refinement** or **block level:** one quick way to characterize the size of a cell or the dimensions of a block, is to say how much times the refinement operations have been performed from the coarsest grid, which is said to be at 0^{th} level of refinement, to the current cell or block. For instance in two-dimensional quadtrees, let $\Delta x^0 \times \Delta y^0$ be the cell size of the coarsest grid. A cell whose level of refinement is n will then be of size $\frac{\Delta x^0}{2^n} \times \frac{\Delta y^0}{2^n}$.
- **Meta-data:** the set of data used to manage the mesh are called meta-data. They can be associated to each cell and contain for example, the level of refinement of the cell, the block it pertains, etc. or they can be more global and contain the connectivities of the grid.
- **Conforming / non conforming mesh:** AMR grids usually look like cartesian ones, or seem to contain some cartesian pieces of mesh. However, at a face separating two cells which are not at the same level of refinement, the corners defining the neighboring cells do not coincide: the mesh is said non-conforming and the **hanging node** is the node that is not shared by the two neighbors. This aspect of the grid may cause some difficulty in finite element approaches, where specific treatment must be provided for the hanging nodes.
- **Grid efficiency:** in patch-based AMR, clustering cells that need to be refined into rectangular patches automatically lead to the refinement of unnecessary cells. The grid efficiency measures then the ratio of the cells that have been flagged for refinement of the total number of refined cells.

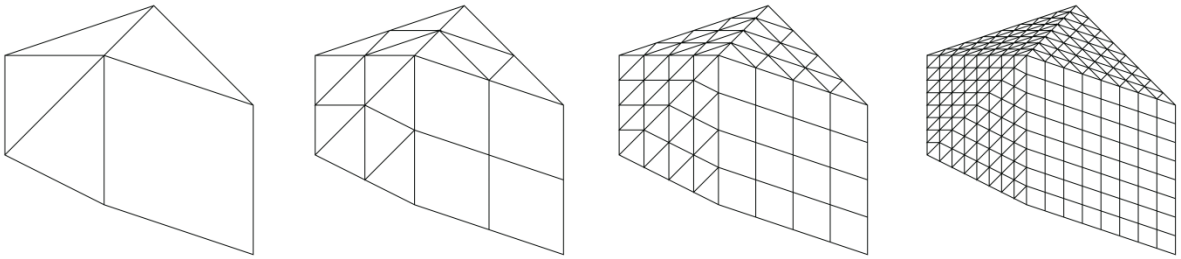


Figure 7.7: *Illustration of the three first levels of refinement of a hierarchical hybrid grid. Reprinted from Hülsemann, Bergen, and Rüde (2003).*

- **Compression rate:** the compression rate may be defined by the ratio:

$$\eta_{\text{comp}} = \frac{N_{\text{unif}} - N_{\text{AMR}}}{N_{\text{unif}}},$$

  where N_{unif} is the number of cells in a uniform grid with a spatial discretization as fine as the finest cells of the AMR grid, and N_{AMR} is the total number of elements (cells, parents cells, ghost cells...) that are required by the AMR strategy. For instance, if the AMR approach is tree-based, then N_{AMR} may comprise all branches of the tree. This ratio may define the gain in memory that AMR methods involve comparing with standard resolution on cartesian grids.

Now, let us present the most widespread AMR methods, which are the patch-based and tree-based AMR approaches. We will see how they are implemented in different AMR codes and the consequences of these implementations on the codes performances.

Other AMR approaches may be found in the literature. Let us just mention here the **hierarchical hybrid grids**, which are based on an initially coarse mesh, made of blocks, each one containing cartesian grids. The connectivity of these blocks do not change through time, but grids they contain can be refined or coarsened independently from the other blocks (see an illustration in Figure 7.7). One can find these methods in the works of Hülsemann, Bergen, and Rüde (2003) for initial unstructured coarse grids, or Golay, Ersoy, Yushchenko, and Sous (2015).

Patch-based AMR

Presentation and definitions

The first method presented here was also the first approach that has been developed. It is called **patch-based** AMR, and involves stacking multiple grids on top of each other in regions where more accuracy is required. The extra grids are always finer than the original grid they are supposed to replace. Once the extra grids are constructed, the equations are solved normally on the finer grids and then interpolated to the coarser ones.

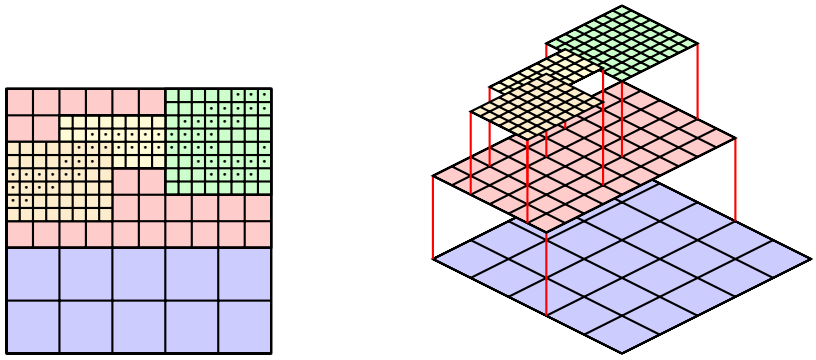
One example of a patch-based grid can be seen in Figure 7.8. The parent-child relationship between the grids is kept using a directed acyclic graph or using linked lists. In the scheme proposed in Berger and Collela (1989), fine grids are allowed to overlap multiple coarser grids and, of course, a coarse grid can contain any number of finer grids. In the first works on patch-based grids Berger and Oliger (1984), finer grids were not necessarily aligned with their parent grids. However, the numerical methods were shown as accurate and much more efficient if they satisfied the following rules for a *properly nested* grid (see Berger and Collela (1989)):

- a fine grid starts and ends at the corner of a cell from the coarser grid;
- two overlapping cells have to be separated by at most one level, called the **2:1 constraint**.

Generating a patch-based grid is made from an initial uniform grid, by clustering cells that have been *flagged* for refinement and by creating a grid with finer cells from this cluster. Clustering algorithms are

one of the key points in the patch-based method. The algorithm from Berger and Rigoutsos (1991) is one of the most widely use, but shows poor parallel performance with distributed computations Wissink, Hysom, and Hornung (2003). A balance between the compression rate and the number of grids must be found. Indeed, the higher the number of fine grids, the more it is complex to manage them, and the more the treatment of boundaries will cost, because the solution there necessitates interpolation procedure. Moreover, to avoid the regridding and clustering operations at each time step, a *safety layer* of fine cells are generated additionally to the patches. For these reasons, the compression rates of patch-based grids are often not as good as those of cell-based grids, that are presented in the next paragraph.

The development of patch-based AMR is related to a refinement strategy based on an estimation of the local scheme truncation error Berger and Oliger (1984). Whenever the domain is asked for re-gridding, the numerical solution, obtained with a finite difference scheme in Berger and Oliger (1984), is computed twice in each cell: once at the current cell size Δx and over two time steps $2\Delta t$, and once over a cell twice bigger $2\Delta x$ and over one time step Δt. The difference between the two solutions gives an idea of the truncation error, according to Richardson's extrapolation Richardson (1911). This difference is then compared with a given threshold, and cells where the error lies below (or above) this threshold can be coarsened (or respectively refined) (see Berger and Oliger (1984); Berger and Collela (1989) for further details). Other strategies rely on the estimation of the local error of the solution, like methods based on an adjoint formulation of the problem (Venditti and Darmofal (2000)), on a posteriori error estimates for *hp* adaptivity (Heuveline and Rannacher (2003)) or a priori estimators (Rannacher and Vexler (2005)) for finite elements methods. Finally, the multiresolution approach enables a mathematically rigorous control of the compression errors (Harten (1994)).



(a) Patches applied to flagged cells (cells marked with a dot).

(b) Grid hierarchy structure.

Figure 7.8: *Example of overlapped grids illustrating patch-based AMR.*

Advantages and drawbacks of patch-based AMR

The main interest of patch-based AMR is the easy implementation of usual schemes for cartesian grids to the cartesian patches. Consequently, some patch-based AMR frameworks can easily be used from existing codes, either for their AMR functionalities, or simply for distributing a uniform Cartesian grids over processing units (MacNeice, Olson, Mobarry, de Fainchtein, and Packer (2000)). Moreover, the resolution over large patches is really efficient, and may counterbalance possible overheads due to the clustering algorithms or inner boundaries treatment.

The distribution over MPI processes is usually made by assigning one process to each patch (but possibly multiple patches per process) as explained in Gunney and Anderson (2016). This helps balancing the work load between the processes, especially when time-stepping methods are used (see discussion hereafter). However, the patches may be of very different sizes, according to the constraints imposed by the method. This size differences make this partitioning step more complicated.

The connectivities between the patches can be defined by the relations of a given patch with the coarser patch it pertains. This type of meta-data can be very heavy, and may not be efficient for large numbers

of refinement levels.

Finally, clustering algorithms may generate overlapping grids. This is avoided as much as possible, but during the evolution of the simulation and the mesh changes, it is frequent that newly created patches overlap the other ones. Overlapping is synonym of duplicated cells, which then deteriorates the method efficiency. To keep an optimal grid hierarchy, the entire re-gridding of the whole domain is regularly necessary, which again increases the computational costs.

Code examples and applications of patch-based methods

Let give below a list of AMR codes that are usually used and referenced. This list of freely available codes is not exhaustive and is inspired from D. Calhoun's website Calhoun (2014).

BoxLib is a widespread block-structured AMR framework, that can be used for solving many different PDE's systems. Several applications are based on BoxLib, like CASTRO (Almgren, Beckner, Bell, Day, Howell, Joggerst, Lijewski, Nonaka, Singer, and Zingale (2010)), MAESTRO (Nonaka, Almgren, Bell, Lijewski, Malone, and Zingale (2010)), Nyx (Almgren, Bell, Lijewski, Lukić, and Andel (2013)) codes.

SAMRAI (Hornung and Kohn (2002); Gunney and Anderson (2016)) is also widely used to solve PDE's in various physical domains.

AMRClaw (Berger and Leveque (1998)) is the *first* AMR version of the finite volume code ClawPack from Leveque (2004). It is based on the works and algorithms of Berger and Collela (1989).

AMROC (Deiterding (2004)) is an MPI-parallel and C++ object oriented code based on the algorithms of Berger and Collela (1989) for solving hyperbolic systems of conservation laws.

Chombo (Adams, Colella, Graves, Johnson, Keen, McCorquodale, Schwartz, Sternberg, and Straalen (2013)) is a software that enables solving multiple PDE's and is able to handle complex geometries (by using an Embedded boundary approach).

Tree-based AMR

Presentation and definitions

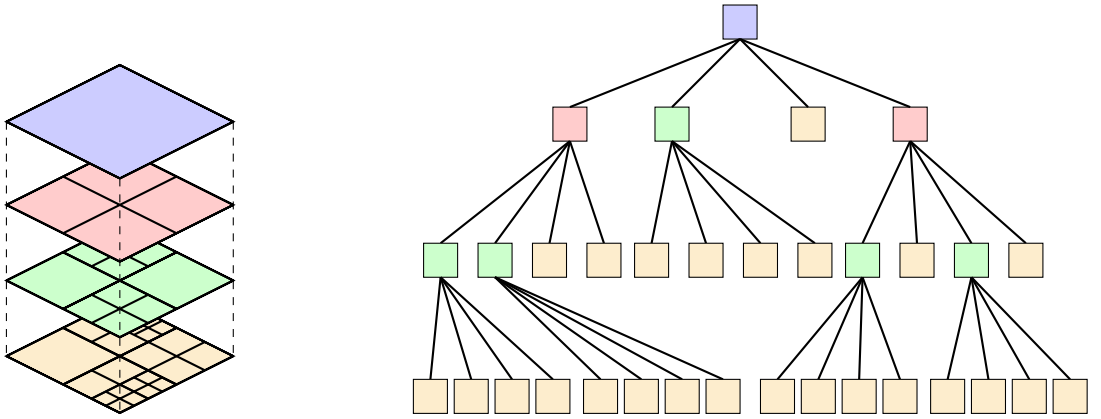
The tree-based method involves modifications of an initial coarse mesh (usually a single cell representing a rectangular domain) by means of recursively dividing its elements into multiple sub-elements with a fixed ratio. These elements may be blocks or individual cells: with individual cell, the method is called **cell-based**. There is non need for clustering algorithms, since each element can be refined or coarsened (with it neighbors) independently of the other elements of the grid. Compared to patch-based AMR, this approach avoids the unnecessary refinement of large-sized zones.

The successive refinement stages lead to a structure of the mesh in the form of a tree. For example in cell-based AMR, cells are refined into two finer cells in 1D, generating a **binary tree**, into 4 new cells in 2D, generating a **quadtree** and into 8 finer cells in 3D: **octree**. Hence, unlike patch-based AMR, where trees are only used to handle the grid hierarchy, tree-based AMR generally makes heavy use of the tree structure to store the mesh, modify it and go through all its cells. However, the use of new data structures implies new difficulties in implementing numerical methods (new integration routines, storage strategies and load balancing techniques need to be developed). An other challenge is also making use of these tools into already existing codes, using their own data structures.

In Figure 7.9a, we can see an example of a 2D quadtree. A quadtree, like any other tree structure, is directed, in the sense there is only one way from one node to another, and does not contain any cycles.

Some of the main elements of a tree – quadtree or octree – are:

- the node on top that is called the **root node**,
- each node can be tied to 0 or 4 (resp. 8 for an octree) other nodes beneath it. This node is called the **parent** and the 4 (resp. 8) nodes beneath it, if they exist, are called **children**. A group of



(a) Example of cell refinement process illustrating cell-based AMR.

(b) Corresponding representation of the domain using a quadtree structure.

Figure 7.9: *Example of a cell-based grid.*

children is sometimes referred to as a **family** and they have **sibling** relationships between them,

- **leaves** are nodes that do not have children.

Issues of tree-based AMR and solutions proposed in the literature

Managing tree-based AMR meta-data raises multiple issues. One of these issues is to know whether we need to store the whole tree or whether we can store its leaves only. Indeed, the tree structure is useful to get the cells connectivities, but finding neighbors of a given cell can lead in the worst cases to the whole tree traversal. An efficient meta-data management would then necessitate as little computation as possible. This issue of meta-data structure has also to be considered with the fact that, in order to get scalable algorithms, data structures and algorithms have to consume as little memory as possible. The replication of the meta-data over each MPI process is also a crucial question in cases of distributed memory computations.

Over time, various implementation choices have been made to deal with these issues. The first (chronologically), and simplest, implementation involved a pointer-based tree representation with all the connectivity information stored explicitly (see for instance Chiang, van Leer, and Powell (1992)). This approach presents heavy memory consumption and is poorly scalable.

A *lighter* alternative was developed by Khokhlov (1998), the **Fully Threaded Tree**. Most of the connectivity is not stored explicitly, but only some useful relations, leading thus to less memory consumption. There, leaves of the tree are stored continuously in memory: they are thus gathered in quads (2D) or *octs* (3D), enabling to use only one pointer from their parent cell to them, and one pointer from the *quad* to its parent cell. Pointers are also used to get the parent cell's neighbors: 4 other pointers in 2D and 6 in 3D. This algorithm was implemented in the RAMSES code Teyssier (2002) or Gerris Popinet (2003) (recently improved to Basilisk Popinet (2015)).

Recent libraries, based on the tree linear storage, in the form of hash-maps or linear arrays, have been preferred (see for instance Sundar, Sampath, Adavani, Davatzikos, and Biros (2007); Sundar, Sampath, and Biros (2008); Ji, Lien, and Yee (2010)). These solutions have proved to use less memory than the Fully Threaded Trees, to have a better cache locality and are easier to parallelize (Ji, Lien, and Yee (2010)): these points are real advantages in the context of HPC. The issues for these new approaches was then the development of highly scalable algorithms (up to 10^5 processes) for generating the mesh, traversing the trees and in particular for the 2:1 balance operation that is really costly.

The p4est library is one example of cell-based AMR that only stores the leaves of the tree. The management of the cells connectivities heavily relies on the z-order space filling curve (also called Lebesgue curve) as we will see in Chapter 7. These space filling curves are present in numerous AMR approaches, but their use is confined to the load partitioning.

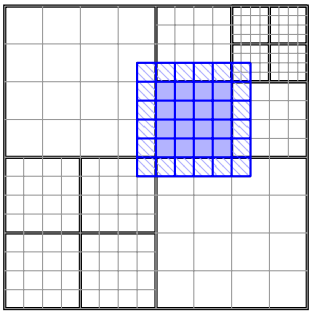


Figure 7.10: *Representation of a* `ForestClaw` *mesh type, with* 8×8 `ClawPack` *grids and an illustration of the ghost cells for the blue grid.*

Code examples and applications of tree-based methods

PARAMESH (MacNeice, Olson, Mobarry, de Fainchtein, and Packer (2000)) is a fully-linked tree-based framework. The leaves of the tree in PARAMESH are made of blocks. This AMR framework is equation-independent, as it only provides routines for the management of the meta-data.

RAMSES (Teyssier (2002)) is a N-body solver and finite volume cell-based code for the resolution of astrophysical problems. The AMR data structure is based on the Fully Threaded Tree of Khokhlov (1998).

Gerris (Popinet (2003)) and Basilisk (Popinet (2015)) are two finite volume codes, also based on the FTT of Khokhlov (1998), and that allow to solve incompressible fluid flows.

p4est (Burstedde, Wilcox, and Ghattas (2011)) is a C cell-based library, relying on Morton's indexing for storing the leaves of trees in a linear array. p4estlibrary also enables to perform simulations over domains of complex geometries, by means of its notion of *forest of trees*. This library will be described with further details in the following section.

deal.II (Bangerth, Hartmann, and Kanschat (2007)) is a C++ library that can be used for the resolution of finite element problems. It initially contained its own cell-based structure, but now interfaces the p4estlibrary Bangerth, Burstedde, Heister, and Kronbichler (2011).

ForestClaw (Burstedde, Calhoun, Mandli, and Terrel (2013)) was developed as an assembly of p4est and ClawPack (see details on ClawPack in Leveque (2004)): each cell of the p4est mesh is the support for a grid of ClawPack. This grid may be of any size, but some works on the performances have shown that a 32×32 grid in 2D enables a real gain of computational time. The cell-based framework is here only used for the coarse grids management and to benefit from p4est good performance on highly parallel computers. Although the increase in performance of this method is proved in comparison with uniform meshes of equivalent cells, it is not obvious that for a same solution accuracy, the larger number of cells created by the grids are really necessary, and whether the grid efficiency is good. However, this hybrid methods presents useful aspects for the implementation of high-order methods with large stencils.

Review of AMR numerical features

Agnostic vs applicative AMR codes:

There exists a large number of patch-based AMR codes. Many of them are dedicated to the resolution of a given physical model or a given numerical method. One can mention here the numerous astrophysical codes, like Ramses, AstroBEAR, Enzo, MAESTRO, FLASH etc. or AMR frameworks dedicated to hyperbolic conservation laws like AMROC, AMRClaw. But several AMR codes have also been developed so that they can be used in a wide range of applications. Among these codes, one has BoxLib, Chombo, or SAMRAI or even PARAMESH which only provides mesh functionalities.. These frameworks are said to be *agnostic*.

Finite volume and finite element methods

Many block-based AMR methods enable to solve hyperbolic systems of equations, using finite volume methods. Indeed, every finite volume or finite difference scheme naturally applies on rectangular blocks of cells. The main difference with Cartesian grids lies at the borders of the blocks, which are part of the inner computational domain: cells at these borders are traditionally updated through the interpolation of the neighboring solution at a finer or coarser level (see Berger and Oliger (1984)).

Although finite element schemes do not specifically require a cell-based type mesh rather than a block-based one, they are associated with some widespread cell-based codes, such as deal.II Bangerth, Hartmann, and Kanschat (2007), and represent the first applications to p4est library (through geophysical simulations Burstedde, Wilcox, and Ghattas (2011); Burstedde, Stadler, Alisic, Wilcox, Tan, Gurnis, and Ghattas (2013)). Finite element methods apply easily on adapted meshes, with some care to be taken at the *hanging nodes*. The solution chosen in Burstedde, Stadler, Alisic, Wilcox, Tan, Gurnis, and Ghattas (2013) to deal with this difficulty is to consider only the *independent* nodes for solving the equations, and interpolate the solution on the hanging ones.

Nature of the equations that may be solved:

Associated with the finite volume methods many AMR methods are well-suited for the resolution of hyperbolic systems of conservation laws: this is the case of AMROC, AMRClaw for example.

However, for many applications, such as incompressible flows, Magnetohydrodynamics (MHD) flows, etc. one needs to solve elliptic equations, like Poisson's equation, which reads:

$$\Delta\Phi = b.$$

Many AMR codes propose then elliptic solvers, as multigrid (see Minion (1996); Almgren, Bell, and Szymczak (1996); Howell and Bell (1997); Sampath, Adavani, Sundar, Lashuk, and Biros (2008)) or multilevel (Popinet (2003)) methods for instance, or are associated with external libraries, such as PETSc (Balay, Abhyankar, Adams, Brown, Brune, Buschelman, Dalcin, Eijkhout, Gropp, Kaushik, Knepley, McInnes, Rupp, Smith, Zampini, Zhang, and Zhang (2016)) or Hypre (Ashby and Falgout (1996)).

Local time stepping:

One other feature often encountered in AMR approaches is local time stepping methods. The idea is to not only save computational costs by reducing the number of grid cells, but also by updating the solutions in these cells at a frequency closer to their own CFL condition. For example, a coarse cell could be updated twice as often as a one-level finer one. Once more, one should take care at the blocks boundaries and use specific time integration methods, such as advancing first the coarser cells and then interpolating temporally at the intermediate time step Berger and Collela (1989) or using Adams-Bashforth method Golay, Ersoy, Yushchenko, and Sous (2015). Local time stepping also requires the knowledge of all cells of the same size to be efficient. This is why it is commonly used in block-based AMR methods, but one can also find it in cell-based approaches, as in Khokhlov (1998).

Parallel computation and scalability:

One required feature for today AMR codes is their good scalability on highly parallel architectures. As already mentioned at the beginning of this introduction, a good work load balance between the processors is essential. To achieve this goal, many AMR codes use space filling curves, like Peano, Hilbert or Lebesgue curves, to go through all the grid cells and then define balanced subdomains associated to each MPI process. The second advantage of these curves is that they preserve the locality of the data: two neighbors on the curve may very probably be neighbors in the real 3D space.

Another tendency today in block-based AMR is the use of OpenMP threads in each block of cells, as in BoxLib or Chombo for example.

Criteria for the choice of an AMR library

To sum up the different aspects that have been presented in this introduction, let us establish a list of criteria, which may help to choose an AMR library for the resolution of our equations. This AMR framework should:

1. provide an interesting compression ratio of the data,
2. scalable on highly parallel architectures,
3. be agnostic (equation independent) and compatible with different structures of the equations,
4. be easy to use even on domains with complex geometries.

The first point directs us rather to tree-based AMR codes with linear storage of the leaves. Moreover, with the perspective of considering multi-resolution schemes, cell-based AMR seems interesting. The third point means that we want the AMR library to only manage the meta-data (the mesh). Finally, all the criteria are fulfilled by the p4est library: the second point has been verified up to 458 000 cores in Isaac, Burstedde, Wilcox, and Ghattas (2015) and the last point is answered by two principles used in p4est that are the inversion of control principle and the forest of trees. In Chapter 7 we present in more details p4est principles and main functionalities. Then in Chapter 8, we present the applicative code CanoP, which uses p4est to solve systems of conservation laws with finite volume schemes.

Chapter 7

Description of the p4est library

p4est library has been developed to serve as a parallel cell-based AMR-grid generator and manager for an increasingly large spectrum of physical applications, since there is a decoupling between the mesh and numerical method. This is then the first interesting point of p4est library. The other attractive points of the library, and probably its main force, lie in its algorithms that were proven to have a very large scalability. p4est moreover handles grids over domains with geometrically complex boundary conditions. Finally, the inversion of control principle (also called Hollywood's principle), makes it very flexible and easy to use.

p4est library makes a heavy use of the z-order space filling curve (and associated Morton's index), so as to store linearly the meta-data associated with the leaves of the tree and partition the work load between the MPI processes, as is done in several other AMR codes. The Morton index is also widely used in the grid adapting algorithms, based on the previous works of Sundar, Sampath, Adavani, Davatzikos, and Biros (2007); Sundar, Sampath, and Biros (2008) for one AMR tree. These algorithms have been extended to a forest of trees in Burstedde, Wilcox, and Ghattas (2011), and have been improved in order to manage efficiently the inter-tree connectivities and to be scalable on parallel architectures (Burstedde, Wilcox, and Ghattas (2011); Isaac, Burstedde, Wilcox, and Ghattas (2015)). The principle of a forest of tree allows to map physical domains with very complex geometries.

While the meta-data of the mesh structure is handled by p4est library, the heavy data or user's data, specific to each application, are left to the user, or can be managed by the SC library, jointly developed with p4est: memory allocation is then done through memory pools. SC library indeed aims at helping some scientific computations, and contains in particular the definition of C containers and associated functions for dynamic array management in a parallel environment.

In this chapter, we are giving more details on the main principles of p4est library, which are:

- the indexing of the forest of trees,
- the scalable algorithms for mesh adaptation,
- the interface with a client code via Hollywood's principle.

7.1 Encoding of a forest of trees by p4est

7.1.1 Macro-mesh and forest of trees

In p4est, a first discretization of a physical space Ω can be represented by multiple trees, called the **forest**, each tree covering a subset Ω_k of the physical domain. This **macro-mesh** can be defined manually by the user, or generated by a meshing software, such as CUBIT or Gmsh. It however has to respect the following rules:

- it is made of four-angled cells in 2D and hexahedral cells in 3D,

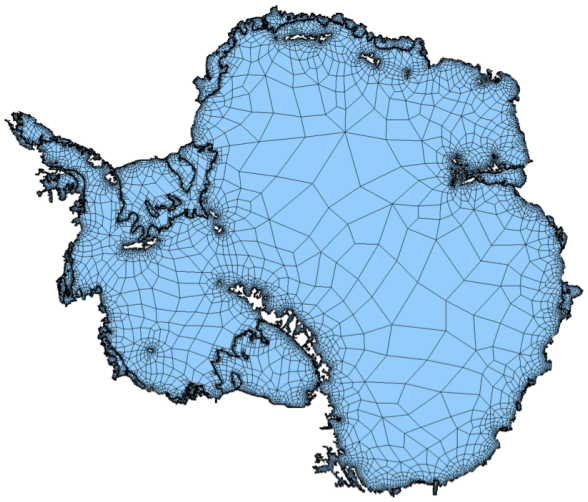


Figure 7.1: *Top-view of the 3D gridding of Antarctica, made of 28 000 octrees. Reprinted from Isaac, Burstedde, Wilcox, and Ghattas (2015).*

- it should be **conforming**.

An illustration of such macro-meshes is given in figure 7.1. In the following, all topological elements will get an index: faces will be named f_i, $i \in \{0, ..., 3\}$ in 2D, $i \in \{0, ..., 5\}$ in 3D ; edges e_i, $i \in \{0, ..., 11\}$ in 3D (they are not defined in 2D) ; corners c_i, $i \in \{0, ..., 3\}$ in 2D, $i \in \{0, ..., 7\}$ in 3D. Their indexing for any leave of a given tree will always be the same, and is given in Figure 7.2. However, two adjacent trees may have different coordinate systems, according to the positions of the corners that define these trees. Thus, the **connectivities** between each of the trees forming the macro-mesh are stored explicitly using some matrices, that specify for instance (see more details in the code documentation):

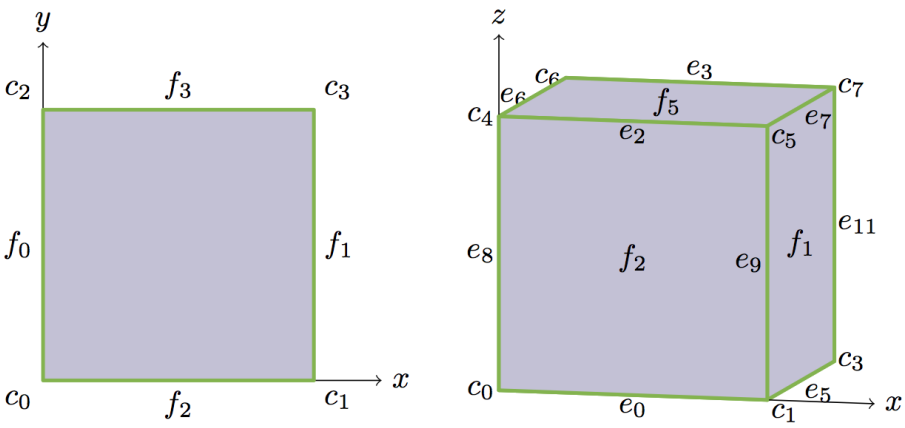
- for a given face of a given tree, who the neighboring tree is,
- for a given face of a given tree, what is the index of the same face in the neighboring tree,
- which corners are shared by multiple trees, and for each of these corners what are the neighboring trees and what is their corresponding corner number

At domain boundaries, the neighboring tree can be referenced as the same inner tree and the same inner face, meaning that the user will impose here a physical boundary condition. Or the neighboring tree can be referenced as an other inner tree (or the same tree with an other inner face) and this will automatically generate periodic boundary conditions.

This connectivity definition allows for a huge flexibility in the macro-mesh definition. For instance, one can easily define a Möbius strip or a Klein's bottle.

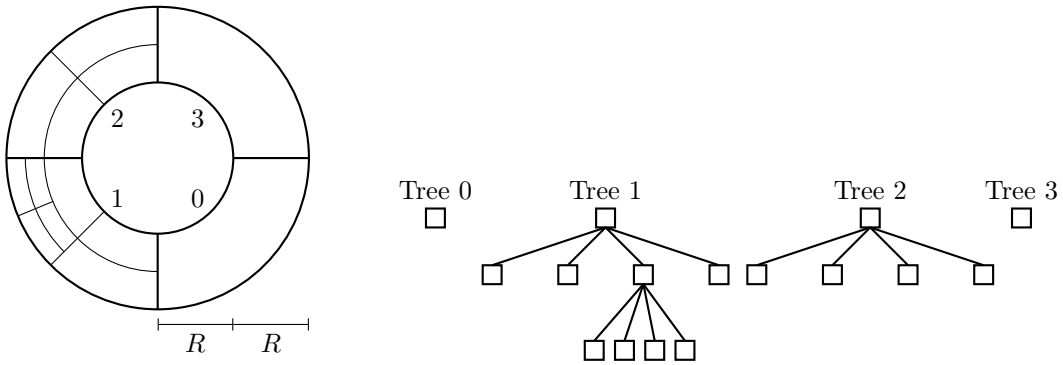
In the connectivity definition, only the corners are linked to the physical coordinates. The remainder is only referenced by the different integer indices. This decoupling enables a proper mesh definition without round-off errors. Moreover, the user can refer to the connectivity matrices at any time and thus recover the physical coordinates of a given point. This macro-mesh will not change during a computation.

Now the macro-mesh is correctly defined, each tree can evolve independently (or almost independently) from the others as the mesh is refined and coarsened. Indeed in p4est, each tree is based on reference cube $[0, 2^b]^d$, where b is the maximum level of refinement and d is the space dimension. Thus, each subdivision of the based cube (named **quadrant** in 2D and **octant** in 3D), which is represented as a leaf in the tree, gets integer spatial coordinates $(x, y, z) \in [\![0, 2^b]\!]^3$. See in Figure 7.3 a simple example of a 2D forest of four trees, with its physical (spatial) and its quadtree representations.



(a) 2D quadrant. **(b)** 3D octant.

Figure 7.2: *Topological elements numbering. Reprinted from Burstedde, Wilcox, and Ghattas (2011).*



(a) Division of an annular domain into 4 macro-cells. **(b)** The corresponding representation of the domain using 4 quadtrees.

Figure 7.3: *Example of a 2D annular AMR grid and its equivalent representation in the form of a simple forest of quadtrees.*

7.1.2 Morton index and linear storage

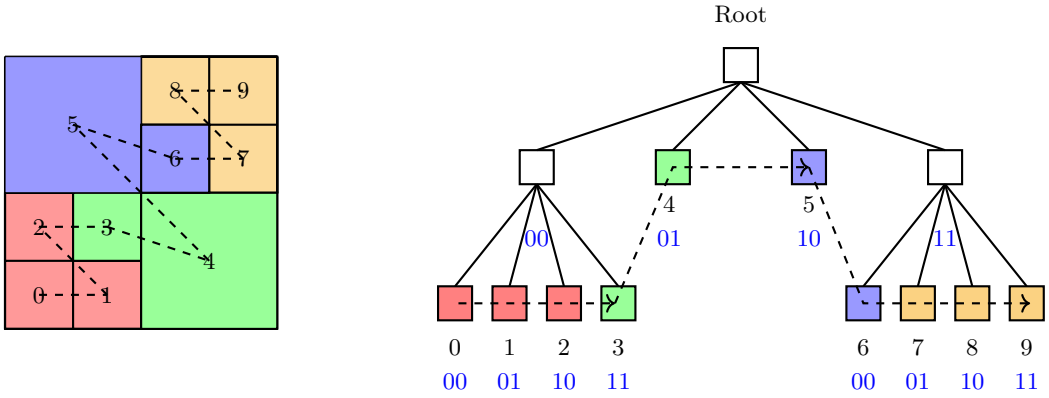
In p4est, the z-order curve and Morton's index (Morton (1966)) are used for many purposes:

- they are used to index all the cells of a grid, as illustrated in dashed line in Figure 7.4a and store their data in a linear array,
- the definition of Morton's index makes it easy to find a neighbor, a parent or a child of a given cell,
- the linear array can be evenly partitioned and distributed over the MPI processes.

One tree binary indexing

The Morton curve is illustrated in dashed line in Figure 7.4a for a spatial grid and in the corresponding tree, in Figure 7.4b: this ordering comes down to just visit the leaves in their standard order. The blue numbers in this figure give the binary index of each node in relation to its family. The combination of all stages of indices up to the leaves is used to order them in the z-curve: for instance, the index of cell number 3 is obtained by taking its parent index $\overline{00}^2$ and adding its own index $\overline{11}^2$ to give $\overline{0011}^2 = 3$, notation $\overline{\cdot}^2$ indicating numbering in base 2.

Finally, let us note that this two-bit binary numbering in 2D corresponds to the position of the quadrants



(a) Adaptively refined square domain. Mesh and z-order curve.

(b) The corresponding representation of the domain using a quadtree.

Figure 7.4: *z-order traversal of the quadrants in one tree of the forest and load partition into four processes. Dashed line: z-order curve. Quadrant label: z-order index. Color: MPI processes.*

in their family, where the origin is the lower left corner. Indeed, quadrant 1 in Figure 7.4a stands at $\overline{x}^2 = 1$ and $\overline{y}^2 = 0$ which gives its index $\overline{01}^2$ relatively to its parent node. The indexing of all the cells contained in the linear array can thus be constructed by considering the binary representations of the coordinates and interwoven in the following way:

$$\overline{m}^2_{3i+2} = \overline{z}^2_i, \quad \overline{m}^2_{3i+1} = \overline{y}^2_i, \quad \overline{m}^2_{3i+0} = \overline{x}^2_i, \quad \forall i \in [\![0, b-1]\!], \tag{7.1}$$

where b is the maximum level of refinement, $\overline{m}^2$ is the sequence of bits constituting the Morton index, $\overline{x}^2_i$ is the i-th bit of the binary representation of x.

We can see that using the z-order curve ensures the locality of memory use: indeed, a neighbor in Morton's index is probably also a neighbor in the 2D or 3D physical space. Cache memory is thus better used, as is shown in Bader and Zenger (2006) for matrix multiplication using Peano curve.

Establishing relationships between cells

Since the methods and algorithms are similar in 2D and 3D, we will talk in the following only about *octrees* and *octants*, designing nodes of the octree.

Using Morton index to order the octants of one tree enables to locate easily the parent of a given octant, its children, its neighbors through a face or an edge, by processing in simple bit flips in the indices. Details on these algorithms are given in Burstedde, Wilcox, and Ghattas (2011). Let us consider only two of them here, given as examples.

The Algorithm 1 takes as input an octant, and returns the coordinates and level of refinement of its parent. This algorithm states that for finding the coordinates of the parent of a given octant o, we are only interested in the meaningful bits representing the parent's level of refinement and higher levels. This is done by "erasing" the value of the bit of weight h, which represents the child's level of refinement. It also comes to dividing by 2 the coordinates in each direction of quadrant o.

Algorithm 1: Get the parent of an octant.

Data: Octant o

1 $h \leftarrow 2^{b-o.l}$;
2 $p.l \leftarrow o.l - 1$;
3 $p.x \leftarrow o.x \ \& \ \neg h$;
4 $p.y \leftarrow o.y \ \& \ \neg h$;
5 $p.z \leftarrow o.z \ \& \ \neg h$;

Result: p

- b is the maximum refinement level.
- For an octant o, $o.l$ denotes its level and $(o.x, o.y, o.z)$ its coordinates.
- & denotes a **binary AND** that gives 1 only if the two bits are also 1.
- | denotes a **binary OR** that gives 1 if any of the two bits is 1.
- $\neg$ denotes a **binary NOT** that gives 1 if the bit was 0, and 0 otherwise.

The Algorithm 2 returns the coordinates of the neighboring octant through a given face. This algorithm simply adds or subtracts the length of the current octant to its coordinates, according to the face number. One can note that the algorithm returns a octant of the same level as the current octant. This octant may not exist: it can indeed be finer or coarser the the current quadrant. Thus, once Algorithm 2 has been called, the effective neighbor has then to be found.

Algorithm 2: Get the neighbor of an octant through a face.

Data: Octant o, Face f

1 $h \leftarrow 2^{b-o.l}$;
2 $p.l \leftarrow o.l$;
3 $p.x \leftarrow o.x + ((f = 0)? -h : (f = 1)? h : 0)$;
4 $p.y \leftarrow o.y + ((f = 2)? -h : (f = 3)? h : 0)$;
5 $p.z \leftarrow o.z + ((f = 4)? -h : (f = 5)? h : 0)$;

Result: p

- b is the maximum refinement level.
- For an octant o, $o.l$ denotes its level and $(o.x, o.y, o.z)$ its coordinates.

Extension to the forest of trees

Up to now, we have presented the ordering of octants in one octree. The specificity of p4est is to be able to deal with a forest of octrees. In the previous paragraph, we have seen the rules imposed to the forest definition, and how the inter-tree connectivities are specified. Here, we present how the Morton ordering of the octants extend to the forest.

This global ordering is simply performed by adding the tree number to the quadrant index, assuming that the trees are defined so as to preserve some locality.

Some algorithms, such as Algorithm 2 may lead to looking for a neighbor octant in a neighboring tree, or at the exterior of the domain. These octants will then be named **exterior** octants. The coordinates of exterior octants are numbers below 0 or above 2^b where we recall that b is the maximum level of refinement. To manage this situation, two additional bits are used in the octant index.

When we are looking for exterior octants, the inter-tree connectivity must then be used to:

1. identify the neighboring octree that contains this octant. This is given by the matrix that relates faces to trees, edges to trees, etc,
2. project to the coordinate system associated with the neighboring octree.

The first point results in one neighboring octree when looking for face neighbors, and in multiple ones when looking for edge or corner neighbors. For these last two cases, the Algorithms 9 and 11 in Burstedde, Wilcox, and Ghattas (2011) enable to find respectively the set of connected octrees.

The second point is performed by the Algorithms 8 for face neighbor, Algorithms 10 and 12 for edge and corner neighbors respectively in Burstedde, Wilcox, and Ghattas (2011), where they are detailed. These algorithms are more complex than the simple bit flips necessary inside one octree, since the two neighboring trees may not be oriented with the same coordinate system.

7.1.3 Partitioning between MPI processes

Let us first assume that we are performing the computation over only one MPI process. The Morton indexing, completed by the tree number and the octant level of refinement, enables a bijective representation of the octree from the linear array to the multidimensional space.

Now, the partitioning of the octants to multiple MPI processes is naturally performed by dividing the linear storage array (see illustration in Figure 7.4 where each color stands for one of the four MPI processes). The partition may be balanced using the division of total number of octants per the number of processes, or according to a weight each octant could be associated with by the user. A good load

balancing is necessary for the processes not to wait too long for others at the communication points. One can note that cutting the Morton curve may lead to MPI subdomains that are sometimes not connected, and whose ratio of boundary surface over inner cells volume is not optimal. However, as we will see in the next chapter, the overhead related to this fact does not prevail.

To facilitate the research of a given octant and determine its **owner** process, further to the call for Algorithm 2 for example, a list of triples (N_p, k_p, o_p) with N_p the number of octants *owned* by the process p, k_p its first local octree referenced in the storage array and o_p the first level-b descendant of the first octant of the array, is transmitted to each process.

The Algorithm 13 of Burstedde, Wilcox, and Ghattas (2011) returns then the owners of the face, edge or corner neighbors of a given octant, by a binary search in the list (N_p, k_p, o_p).

This algorithm is called whenever the user is looking for a neighboring octant that does not lie in the same MPI process as its current one, as well as in the algorithms performing the 2:1 balance or creating the ghost layers.

7.2 Algorithms for the creation and adaption of a mesh

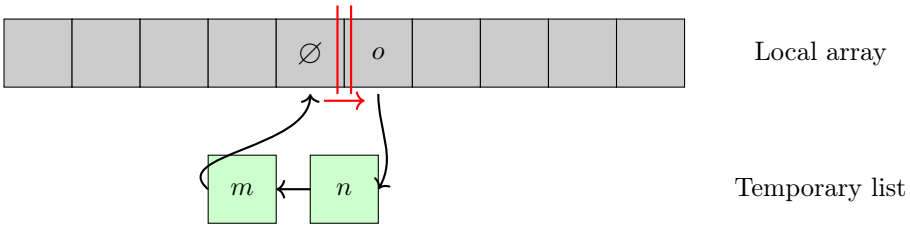
Let us choose a macro-mesh, a maximum level of refinement and a refinement criterion, we can then call p4est callback functions to proceed in the forest of trees creation, the refinement and coarsening of a selection of cells. Let us recall that all these algorithms apply on the meta-data (related to the tree structure and the mesh management).

7.2.1 Creating the forest and initializing the mesh

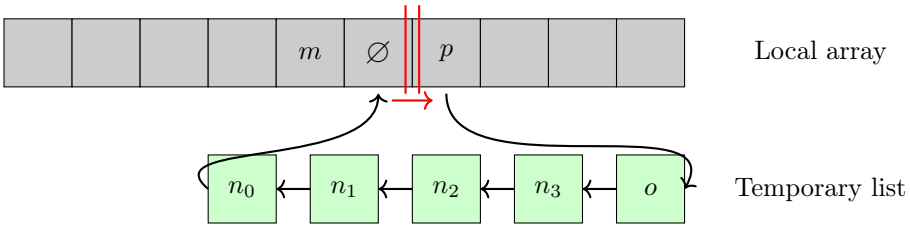
Mesh creation is done once, at the very beginning of the simulation. If the initial solution is not imported from a file (resulting from the dump of a previous solution), thus in each tree, uniform meshes with cells of a given level are generated. This initial level of refinement is specified by the user. Then, p4est also creates the first linear arrays containing all these cells, indexed by the Morton index as described above, and distributed over the processes. Finally, the user can refine and coarsen the mesh recursively, according to the refinement criterion and based on the initial solution or a replacement function defined by the user.

7.2.2 Refining algorithm

The refinement of the mesh can be done each time the user needs it. It is performed by one traversal of the tree, each cell being then refined by one level, or recursively refined until the refinement criteria are satisfied. Refining the mesh leads to add cells to it, thus to increase the size of the storage arrays. This is done locally by each MPI process, as there is a priori no need for inter-process communication. Figure 7.5 illustrates the process of refinement as performed in p4est, for the non-recursive mode and in a quadtree (2D). This procedure preserves the ordering of the quadrants, by means of a temporary list that is created and that stores the quadrants waiting for space in the array. This list increases each time a quadrant is refined. Each quadrant of the initial array is appended to it, and leaves space in the initial array. If this first quadrant of the temporary list is not to be refined, then it fills the free space. In the case when it has to be refined, its children are prepended to the temporary list, and the first of them can now fill the free space in the array. The refinement algorithm runs in $\mathcal{O}(N_p)$ (recall that N_p is the local number of octants on an MPI process).

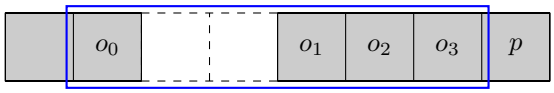


(a) Quadrant m is being put back in the final array while quadrant o is appended to the temporary list.

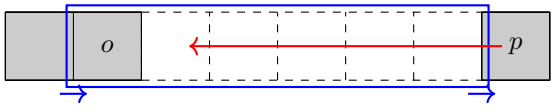


(b) Quadrant n has been refined and is replaced by its 4 children $n_i, i \in \{0, .., 3\}$. Quadrant n_0 is being put back in the final array while quadrant p is appended to the temporary list.

Figure 7.5: *Two steps of the array traversal for refining the quads.*



(a) Successive quadrants $o_i, i \in \{0, ..., 3\}$ is being tested whether it is a family and whether it must be coarsened.



(b) Family $o_i, i \in \{0, ..., 3\}$ has been coarsened and is replaced by its parent o. The sliding window made of empty spaces moves then one step forward, and quadrant p is copied at its beginning.

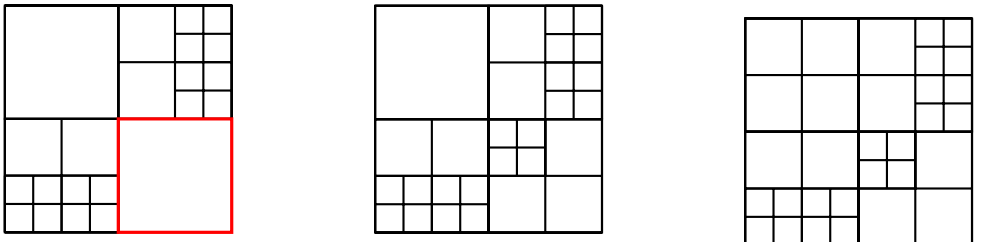
Figure 7.6: *Two steps of the array traversal for coarsening the quads.*

7.2.3 Coarsening algorithm

Such as the refinement algorithm, the coarsening one is able to modify the octant array while keeping the Morton's ordering in only one traversal of the local array. The algorithm makes use of a **window** that surrounds the empty space, which is created by the coarsening of the octants. This is illustrated in Figure 7.6 for the case of quadrants (2D), where, in the first step, the group of quadrants $o_i, i \in \{0, ..3\}$ is tested for being a family (getting the same parent) and needing to be coarsened. If this condition is true, then the family is replaced by its parent, leaving space in the initial array. The window then moves one step forward and the first quadrant next to the empty zone is copied at the beginning of the window. This procedure also runs in $\mathcal{O}(N_p)$.

7.2.4 2:1 balancing algorithms

The 2:1 balance property of a mesh means that two neighboring cells can not be more than twice as large or more than half as small. This property is required for some numerical schemes, for example in finite



(a) Mesh not satisfying the 2:1 balance: the red cell is neighboring four times smaller cells.

(b) Mesh satisfying the 2:1 balance through faces, but not through corners.

(c) Mesh satisfying the 2:1 balance through faces and corners.

Figure 7.7: *Cell based meshes that do and do not satisfy the 2:1 balance constraint.*

element methods, and is considered as being better in general to get smooth transitions from coarse to fine cells. The balance can be verified across faces only, or across faces, edges and corners (see Figure 7.7 for a comparison of the different versions and an example of unbalanced mesh).

The algorithms for 2:1 balancing are called after a mesh modification (refining and coarsening). They are among the costliest algorithms in cell-based methods, especially when executed in a parallel environment. Much work has been done to improve their execution times and their scalability (see the latest improvements for p4est in Isaac, Burstedde, and Ghattas (2012)). We will not give the details of these algorithms: they can be found in Sundar, Sampath, and Biros (2008); Burstedde, Wilcox, and Ghattas (2011); Isaac, Burstedde, and Ghattas (2012). The main idea to maintain a good computing performance, is to avoid the number and the size of the messages that are passing from one process to the others. Indeed, the 2:1 balance process needs information on the neighbors of octants, and very often, if one octant is to be modified (refined), then some of its neighbors have to modified too. The balancing involves then a recursive procedure. If a naive procedure is used, meaning without accounting for the distributed aspect of the data, 2:1 balance can be very costly in terms of MPI communication.

In p4est, the procedure for 2:1 balancing requires only one inter-process exchange, by using the notion of **insulation layer** of an octant. The insulation layer of octant o, of level l is the set of octants of same size l that surrounds o. These octants may not *exist*: their parent or children may replace them in practice. In these cases, we however say that these octants are contained in o's insulation layer. If one of the octants in o's insulation layer is very small, meaning, it has a level of refinement higher than $l+1$, then it can cause an unbalanced situation. However, this must finally be checked by o's owner, so as to avoid useless refinement. The rule to restore the 2:1 balance is indeed to refine the minimum necessary octants.

2:1 algorithm of p4est is then split into four steps:

1. each MPI process performs a 2:1 balancing for its local octants,
2. for each octant o for which a part of the insulation layer lies out of the set of local octants, a message is sent to the corresponding owners of these exterior octants with the coordinates and level of o,
3. each process that receives a requirement checks within its own octants if one of them could cause o to be split, and if it does, sends it back to o's owner,
4. from this information, o's owner process can finalize the 2:1 balancing among its octants.

7.2.5 Ghost cells

In computations using multiple processes, the spatial domain is split and each process treats a given subdomain. At the boundaries of these subdomains, however, one will often need to access information pertaining to an other process, so as to compute the numerical fluxes for instance. To do so, p4est permits to build a ghost layer, that can be updated each time the user calls the function `p4est_ghost_exchange_data` (or other equivalent functions).

This ghost layer is built by p4est by checking the face and corner neighbors of each cell of the subdomain, and to which MPI process these neighbors pertain (through the `Find_owners` algorithm in Burstedde, Wilcox, and Ghattas (2011) and the use of the z-order curve). If the process is not the local one, the cell's data is added to a buffer array, which is intended to the neighbor's owner. Buffers are then sent between each process.

In a recent improvement of this procedure, as described in Isaac, Burstedde, Wilcox, and Ghattas (2015), ghost may also be determined on grids that are not 2:1 balanced.

7.2.6 Other specific algorithms

Some other algorithms are presented in Burstedde, Wilcox, and Ghattas (2011) that show a high scalability. The first one aims at numbering all the nodes of the mesh, which is a bit more complex than cell numbering. As the use of this algorithm is more specific to finite element approaches, we will not detail it here.

Finally, a checksum function, based on Adler 32 checksum algorithm, can be used to give the forest a highly probable unique identifier. This identifier is computed locally on each process and then combined on the master process through an `All_Gather` call. This function may be used during codes development to ensure non-regression of the code.

7.3 p4est library functions

From a user's point of view, the functions of p4est can be called to create and adapt the mesh, to traverse it, to create the ghost layers and to write the data in output files. For each of this functionality, the user may provide a **callback function**, which tells p4est what has to be done (see a schematic representation in Figure 7.8). For example, when traversing the mesh to update the numerical solution, the user calls `p4est_iterate` function and provides it the callback function related to the solution's update. Thus p4est routines calls for the user's function, according to the Hollywood's principle. A full example of p4est usage can be found in Burstedde, Wilcox, and Isaac (2014). We describe here only some of them, as examples.

7.3.1 Iterators

The function `p4est_iterate`, whose signature is given in Code 7.1, executes a user-supplied callback function: `iter_volume`, `iter_face` or `iter_corner` at each cell, face or corner respectively. The prototype of such callback function is given in Code 7.2: the user gets information about the local quadrant through the pointer `info` and determines what must be applied to that quadrant.

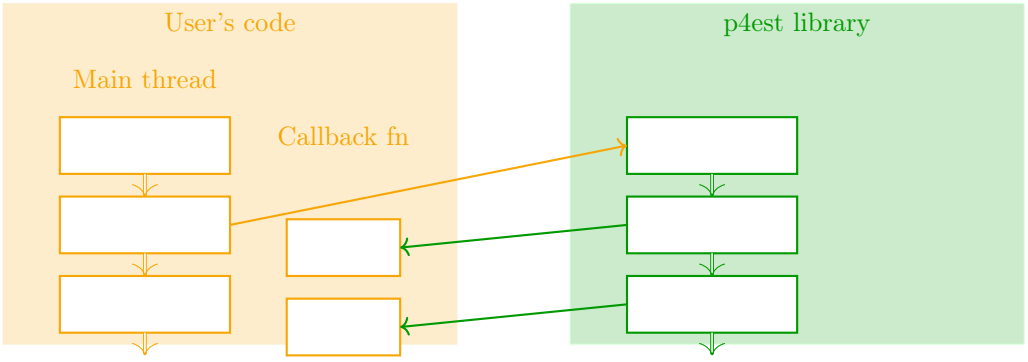


Figure 7.8: *Sketch of the interactions between the user's code and p4est library. Simple arrows stand for the calls of one function to an other. Double arrows stand for the progression of the computation.*

When `p4est_iterate` is called, the grid is traversed according to the Morton's index. The traversal by faces and corners is performed by the traversal by cells and then loops over local faces and corners. In each cell (or at each face or each corner), the callback function is then called and executed.

The structure `p4est_iter_volume_info_t` contains information about the local quadrant, that is given by p4est to the user. Among these information stand a pointer to the quadrant, its identity number, the tree number it belongs to.

Code 7.1: Iterator function signature

```
void p4est_iterate (
        p4est_t * p4est,                        /* p4est forest */
        p4est_ghost_t * ghost_layer,            /* ghost cells */
        void * user_data,                       /* optional argument to user's
                                                callback function */
        p4est_iter_volume_t iter_volume,        /* quadrant callback function */
        p4est_iter_face_t iter_face,            /* face callback function */
        p4est_iter_corner_t iter_corner         /* corner callback function */
);
```

Code 7.2: Prototype of quadrant callback function

```
typedef void (*p4est_iter_volume_t) (
        p4est_iter_volume_info_t * info,/* information about a quadrant
                                           provided by p4est to the user */
        void * user_data                /* the user context passed to p4est_iterate() */
);
```

7.3.2 Refining and coarsening

Refining, coarsening, 2:1-balancing and load partitioning are also p4est functions that require user-defined callback functions. For refinement and coarsening, two callback functions are necessary. The first one, `refine_fn` must return 1 if a given quadrant has to be refined and 0 if not. If the quadrant is to be refined, then p4est uses either the callback function related to initial data filling `init_fn` or the *replace* callback function `replace_fn`. `init_fn` is used if the user decides to fill the quadrant with an analytical solution. This may occur when the initial condition is set. Otherwise, `replace_fn` defines the operations to set the values of the heavy data during refinement: for example, projecting the solution from the old coarse cell into the newly created ones; or during coarsening. The prototype of `refine_fn` is the same whether the grid is refined or coarsened. However, information is provided to the user about the old octants and the new ones, so that it can determines which operation is occurring. The 2:1 balancing is similar, but does not need the `refine_fn` callback function. However, the user can define whether the 2:1 constraint must also be applied through corners or only through faces or edges (in 3D).

7.3.3 Ghost layer creation

A simple call to `p4est_ghost_new` makes p4est create the ghost cells that lie at each MPI subdomain boundary. Then, calling `p4est_ghost_exchange_data` will update the ghost layer and the cells content through MPI communications from one subdomain to the other.

7.3.4 I/O functions

p4est enables to output all the data (user's data and mesh data) in VTK format. Another useful functionality lies in the possibility to dump the whole data, tree structure and MPI partition with the

`p4est_save_ext` function. The filename that is such produced can then be read by `p4est` when calling `p4est_load_ext`, which will re-create the exact forest structure that has been dumped, while the MPI partition can be changed (by changing the number of MPI processes for instance).

Finally, the function `p4est_connectivity_read_inp` is used to create a new connectivity from an external ABAQUS type file.

7.4 Conclusion, references and current improvements of p4est

We have presented the main principles of `p4est` library. In this library, the z-order space-filling curve and Morton's index are not only used for load partition between the MPI processes, as is done in many other codes, but also for their properties that enable to use simple algorithms to manage the mesh inside one octree. Moreover, these approaches are extended to forest of octrees and allow thus to map complex spatial domains. Scalable algorithms have been developed in Burstedde, Wilcox, and Ghattas (2011) and improved in the subsequent works Isaac, Burstedde, and Ghattas (2012); Isaac, Burstedde, Wilcox, and Ghattas (2015). It is not necessary for a user to perfectly know the details of these algorithms. Indeed, he can use `p4est` library functions, like the iterators, the refinement and coarsening functions, by only designing the callback functions, applied by `p4est` library while traversing the gird, refining or coarsening it. However `p4est` library only manages the meta-data associated with the mesh structure, and no application is associated with the library.

Chapter 8

An AMR applicative code: CanoP

8.1 General presentation of CanoP

CanoP can be considered as an applicative software layer of the p4est library, that enables to solve systems of balance laws with finite volume schemes on 2D and 3D adaptive grids. Indeed, the p4est library asks for the custom code to provide callback functions. These callback functions are defined in CanoP, within a framework of classic finite volume solvers, meaning that iterations are performed over the cells to compute fluxes at the cells interfaces and then updating the values of the variables inside each cell. CanoP framework aims then at facilitating the integration of new finite volume applications, or new finite volume features, such as high-order schemes, low-Mach number solvers, etc, that have not yet been tested in an AMR context. For these reasons, CanoP code itself is permanently evolving. We are presenting below some features and performance tests of the current version of CanoP, and perspectives for future developments.

Although functionalities provided by CanoP are still basic, it is today one of the first applicative codes for finite volume simulations on AMR grids using p4est. Among such other codes (see also Chapter 7), one can mention ForestClaw, also based on p4est and using Clawpack routines. Clawpack is a library providing a collection of finite volume methods for systems of conservation laws (Leveque (2004)). This code is also under development Burstedde, Calhoun, Mandli, and Terrel (2013). The deal.II library, can also be mentioned here, although it provides finite-element and Discontinuous Galerkin methods. For a few years indeed, the AMR part of deal.II is also based on p4est (Bangerth, Burstedde, Heister, and Kronbichler (2011)).

Brief history: The main structure of CanoP was developed in C language, in 2014, during Alex Fikl's master's internship Fikl (2014). During CEMRACS 2014 summer school, the first applications of CanoP, a scalar advection solver and Suliciu's solver for the 3-equation system (see Chapter 4), were tested as well as the use-friendliness of the p4est library Drui, Fikl, Kestener, Kokh, Larat, Chenadec, and Massot (2016). Later, a solver for the Euler's equations with gravity source term, and a kinetic solver for polydisperse spray models have been added (Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016)). Difficulties have then arisen about the code maintenance and readability due to the integration of these multiple applications. From that moment, an effort has been done to split the different applications and to simplify the process of integrating new ones, through a refactoring of the code using C++ classes, templates and factories.

8.1.1 Features in CanoP:

Structural features: In Figure 8.1, we show a sketch of the general structure of CanoP. On the left-hand side are some of the common functionalities, while on the right-hand side are some functionalities that are specific to each application. Among the common functionalities, one can find:

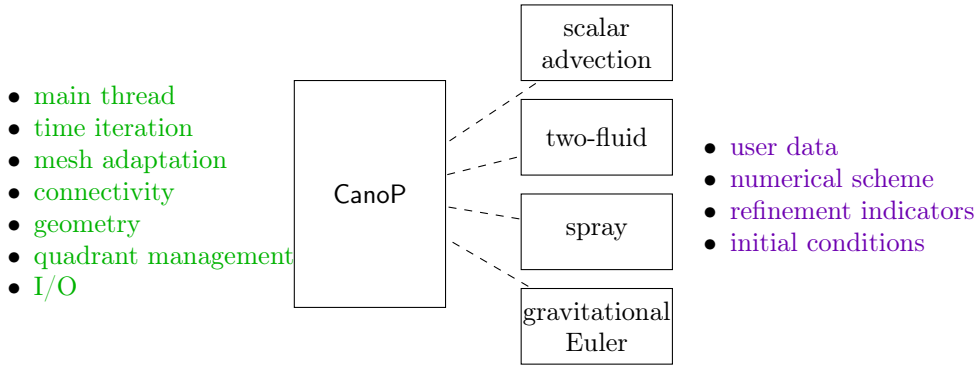


Figure 8.1: *General sketch of CanoP framework and main functionalities: related to the common structures of the code (on the left), that must be defined by each application (on the right).*

- the main steps of the simulation: initialization, loop over time steps, finalization (see also Figure 8.2); bases for finite volume approaches: functions and functionalities are available to help compute time steps, gradients, fluxes...; and AMR tools: routines to automatically handle the mesh adaptation, the work load partitioning between the MPI processes, routines that handle the grid geometries, connectivities...
- another set of functions and functionalities that can be used by the different applications, such as predefined mesh connectivities, geometries or initial conditions.
- input, output and code profiling possibilities.

It is then left to each application:

- the definition of the application variables and associated functions (getter, setter, ...)
- the numerical scheme functions: time step computation, flux computation, cells update... (that are part of the *solver* box in Figure 8.2)
- initial and boundary conditions, as they involve specifically the user's data,
- indicators for refinement and coarsening.

The four current applications are also mentioned in Figure 8.1.

In Figure 8.2 to Figure 8.6 are represented some of the main procedures of CanoP, and their call to p4est routines and to other libraries (Lua, HDF5).

The main stages of a CanoP run are illustrated in Figure 8.2. They compose a classic finite volume procedure, with a domain decomposition using MPI and specific calls to AMR routines (for mesh initialization and mesh adaptation). Otherwise, it consists mainly in an initialization step, where all objects are instantiated, iterations up to a given stopping criterion and a step of finalization destroying all created objects and pointers.

The initialization step is detailed in Figure 8.3. We first recall that p4est provides the possibility to dump the whole data (user's and p4est data) of a simulation, and then reload these data when restarting a simulation. These options are available in CanoP too. At initialization, either a previously dumped solution can be loaded by p4est, or it has to create a new one. In the last case, first a macro-mesh has to be created (named *connectivitiy* in Figure 8.3), then based on this macro-mesh, p4est creates its forest structure and the initial mesh, that is uniform. This initial mesh can then be adapted to the initial solution. In CanoP, we have chosen to set the initial mesh at the finest grid level, and then coarsen it recursively $N = L_{max} - L_{min}$ times.

Mesh adaptation occurs in CanoP at each time step. This choice is probably not the best, since it is computationally expensive and the numerical solution may probably not evolve much from one iteration to the other. However, up to now, we have not tested other options. The mesh adaptation procedure is illustrated in Figure 8.4 and follows five steps:

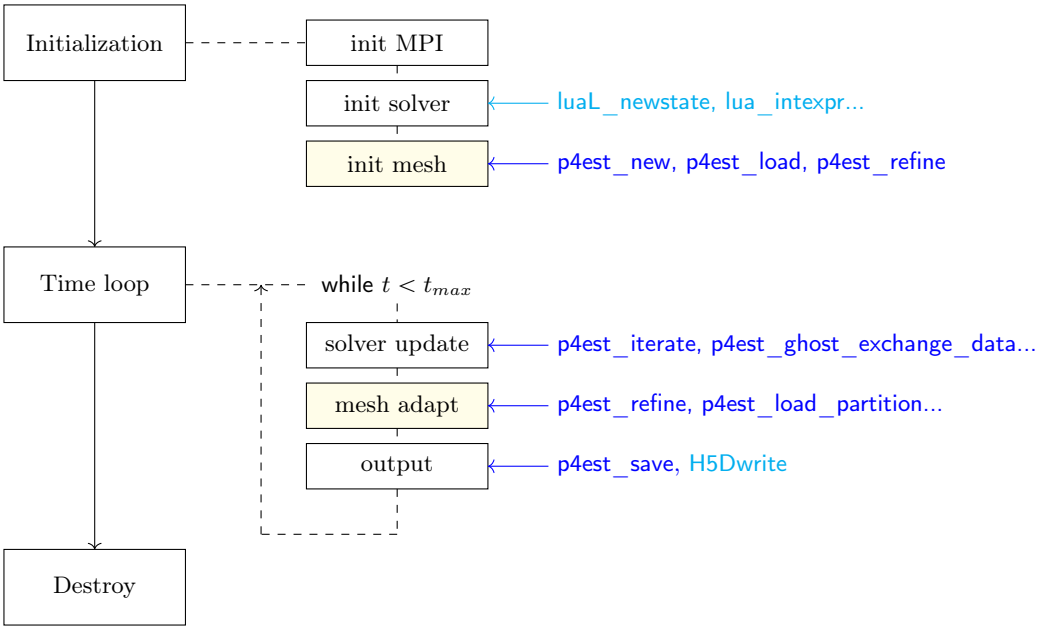


Figure 8.2: *Sketch of CanoP code structure and calls for p4est functions*

1. all cells of the mesh are visited and flagged for being refined or coarsened, according to the refinement criterion and the refinement thresholds that are provided by the parameters of the simulation,
2. the mesh then undergoes a first modification that is the refinement step. This step involves two callback functions: one managing the flags (see Figure 8.5) and one setting the values in the newly created cells (see Figure 8.6).
3. the second mesh change is the coarsening stage. Again, two callback functions are necessary: one managing the flags and that is specific of the coarsening step, and one setting the data in the new cells. This last function is in fact the same as for the refinement step (see Figure 8.6).
4. the last mesh modification ensures a 2:1 balance. Once more, for this step, the replace callback function is used by p4est to set the values in the new mesh.
5. the mesh adaptation procedure ends with a re-partitioning of work load the MPI processes.

The replace callback function illustrated in Figure 8.6 involves a basic option for setting the values in the newly created cells. Indeed, when cells are refined, the variables values of the old cell are simply replicated in its children. When a family of cells has to be coarsened, then an average on some of the variables old values is used to set the new ones in the new cells.

Technical features: We present here some features of CanoP code, concerning choices that have been made to deal with the I/O data.

In CanoP, the parameters of a simulation (see examples of the different options in the table below) are input data provided in a settings file written in Lua language (if a parameter is absent from the configuration file, a default value is then assigned to it). The advantages of using Lua library are its simplicity and flexibility for writing the settings files. Moreover, these files may also contain small functions (written in Lua) that can be used to set functions in the main code, without having to compile the whole code again. This could be used to define new initial or boundary conditions for instance. This is not yet integrated in CanoP.

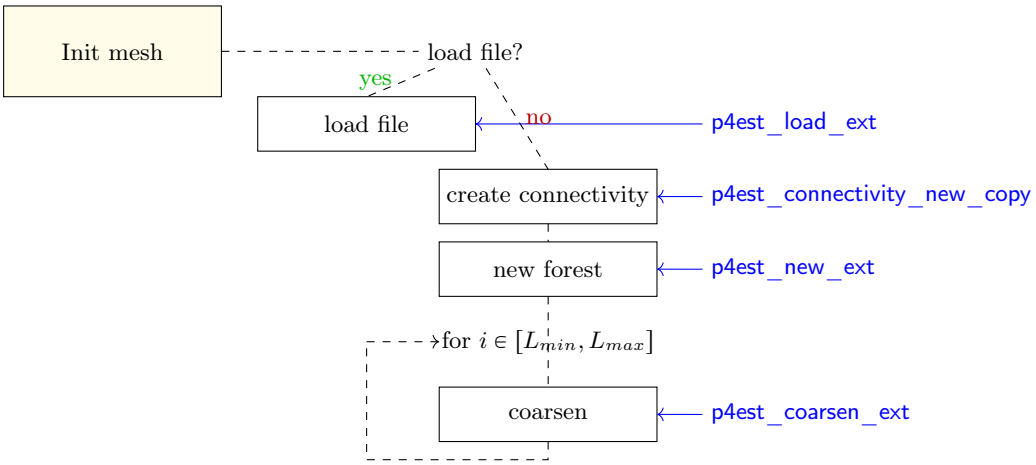


Figure 8.3: *Zoom in the init part structure and calls for p4est functions*

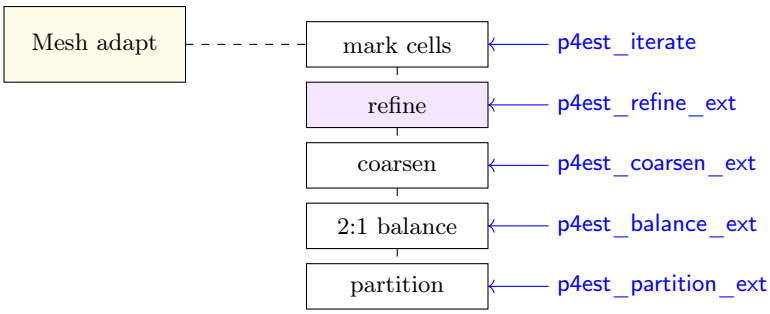


Figure 8.4: *Zoom in the mesh adapt part structure and calls for p4est functions*

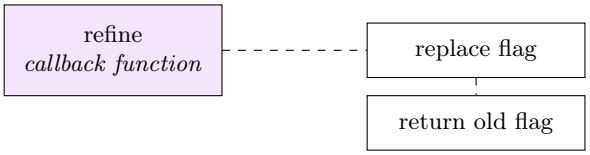


Figure 8.5: *Zoom in the refine callback function, that informs p4est if the cell should be refined.*

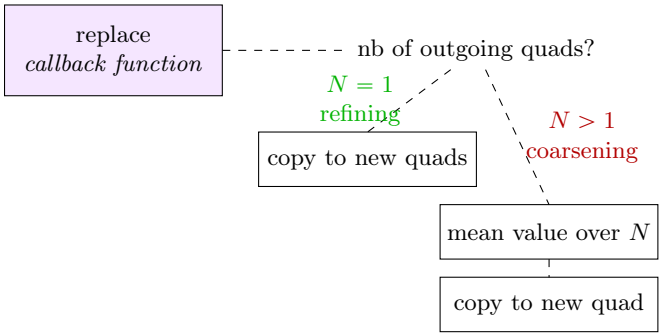


Figure 8.6: *Zoom in the replace callback function, that computes the value of the newly created quadrants.*

Name	Description	Example
`solver_name`	solver to be created by the factory	bifluid
`tmax`	simulation final time	1.0
`cfl`	CFL coefficient	0.5
`min_refine`	minimum level of refinement	3
`max_refine`	maximum level of refinement	10
`indicator`	type of refinement criterion	"alpha"
`epsilon_refine`	threshold for refinement	$2.0 \cdot 10^{-5}$
`epsilon_coarsen`	threshold for coarsening (lower than for refinement)	$1.0 \cdot 10^{-5}$
`connectivity`	type of domain connectivity	"unit"
`geometry`	type of domain geometry (cartesian, cylindrical)	"cartesian"
`initial_condition`	type of initial condition	"bifluid_advection"
`boundary`	type of domain boundary condition	"reflective"
`output_name`	base name of the output files	"advection-3-10"
`save_count`	number of output files	10
`single_precision`	possibility to output data in single precision	1
`mesh_info`	possibility to output mesh information (MPI rank, tree number...)	1
`write_variables`	names of the variables to be output	"alpha, velocity, pressure"
`dump_option`	possibility to dump the whole p4est and users data (to restart the simulation later)	1
`load_solution`	load a previously dumped file instead of creating a new forest	0
`statistics`	possibility to compute code performance statistics	0

One specificity of p4est is its ability to read external mesh files, of type `*.inp` (ABAQUS type files), to create complex connectivities and define the macro-mesh. While reading such mesh files is usual in codes based on fixed unstructured grids, few AMR codes enable it. It is one of the advantages of p4est its concept of forest of tree. CanoP uses this feature of p4est: whether an external ABAQUS file has to be read, its name must be provided in the settings file, through the variable `connectivity`. A small tutorial on the creation of such mesh files using Gmsh is given in Appendix E.

Although p4est offers a possibility to output the users data and mesh information, CanoP has been using its own output functions from the beginning of its development. Among the libraries offering parallel output writing, we have chosen to use HDF5library, which is based on MPI-IO: each MPI process outputs its part of the mesh and the data associated with. These data are made of metadata, provided by p4est, and that are the cells connectivities, and of the user's data (the numerical solution of his simulation for instance). For these files to be read by Paraview, XDMF files are associated with the HDF5files.

Finally, a structure called *statistics* provides a set of time measurements for different parts of the code. This helps drawing a first coarse profiling of a simulation run. When the option is activated in the settings file, these measures are performed throughout the computation, and gathered in an external text file. Moreover, for test cases, for which an analytical exact solution is available, L^1 and $L2$ errors

are computed at the end of the simulation. This is for instance the case with the advection cases, as we will see in the next chapter.

Code development framework: From the beginning CanoP sources have been put on Git repositories, whose access was provided to each team involved in the code development (at Maison de la Simulation, CentraleSupélec and IFPEn). Git also provides some tools (for instance Git hooks) that can be used for code quality purposes. We consider using these tools in the future. The project aims indeed at being available to a larger community, once CanoP framework is better structured so as to accept new applications. Moreover, while a wiki page helps the compilation of the code on several clusters, there is no user's guide yet.

About compilation, let us just mention that CMake can be used to generate the Makefile, as CMakeLists.txt files are available in the project.

Finally, some non-regression tests are being integrated, by using CTest. These tests can be run after each code change and compilation, so as to verify that the latest modifications do not prevent the code from running properly. These tests have the form of short simulations, whose results are compared with reference results. This is useful when different applications are integrated at the same time by different users.

8.1.2 Applications in CanoP:

Let us quickly present here the four applications that are currently integrated in CanoP's framework.

Scalar advection equation:

The application solving a scalar c transport equation at velocity $\mathbf{u}$, using an upwind scheme was the first application of CanoP. The initial value problem simply writes:

$$\begin{cases} \partial_t c + \mathbf{u}^T \boldsymbol{\nabla} c = 0, \\ c(0, \mathbf{x}) = c_0(\mathbf{x}), \end{cases} \tag{8.1}$$

The velocity field is an external parameter and different test cases have been developed with the solver, using different velocity fields.

The scheme that is used is an upwind discretization. The refinement criterion is based on the gradient of the scalar c. Numerical results can be found in Fikl (2014), where tests on subcycling have also been done.

Although basic, this application has permitted to draw the main framework of CanoP code. More features have then been added, according to the new applications needs.

Two-phase models:

The second *historical* equations were those of the two-phase 3-equation model. This system is solved with Relax-S, Relax-Band HLLC scheme (see their definition in Chapter 4), at first and second-order in space and time. The integration of this second model has lead to considering a set of numerical features, presented in Chapter 6. It has also driven to the development of a triangular version of CanoP has presented at the end of the present chapter.

Up to now, only the 3-equation model has been integrated in CanoP, but the integration of the other models derived in Chapter 1, using the numerical schemes of Chapter 4 and Chapter 5 should be straightforward.

Spray models:

A Eulerian high-order moment model for sprays of drops is also integrated in CanoP. It is based on the fractional moments of a number density function, that describes the distribution in size of the

population of drops (see the equations given in Introduction). The equations for the moments are made of a transport part and an evaporation source term part. The equation on velocity is pressureless and contains evaporation and drag source terms. Details on the model and its derivation can be found in Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016); Essadki, de Chaisemartin, Laurent, and Massot (2016); Essadki (2017).

The transport part discretization used for this pressureless system of equations is a kinetic scheme originally developed in Bouchut, Jin, and Li (2003) and then adapted to Eulerian moments model in de Chaisemartin (2009); Kah, Laurent, Massot, and Jay (2012). A specific ODE solver is used to integrate the evaporation source terms Massot, Laurent, Kah, and de Chaisemartin (2010b) and is coupled with the integration of the drag force in the velocity equation.

First results involving the spray model implemented in CanoP can be found in Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016). One of the particularity of the spray model comparing with the two-fluid model is the computational cost of the source terms integration, which is largely prevailing over any other part of the scheme. Indeed, this integration involves the reconstruction of the Number Density Function, while ensuring the realizability of its moments (see in Introduction, section 0.3.3.2 for more details on the moments model). Consequently, in spatial zones where the number of drops is null, updating the variables is immediate: the work load is then no longer defined by the number of cells only, but a weight should be allocated according to the cost of the ODE integration. The computational constraint on the source term integration has lead to consider using heterogeneous architectures and GPUs, in order to reduce the time spent in this task (see Essadki, Jung, Larat, Pelletier, and Perrier (2017)).

Hydrostatic Euler equations

The equations and numerical schemes used in this application are the same as in Ramses code Teyssier (2002). Ramses code is an tree-based AMR code for astrophysical applications, which is currently widely used at CEA.

Thus, with this application, Euler equations with a gravity field are solved using a HLLC scheme for the convective part and an elliptic solver for the gravity source term. The Euler equations are close to the two-phase system, but this application presents a few different features:

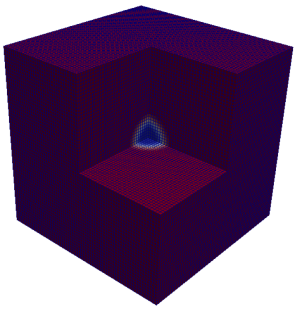
- a version of the solver in cylindrical coordinates is available,
- the full gravitational system of equations necessitates to solve a Poisson's equation for the integration of the gravity source term, that writes as a potential function.

The Poisson's solver is currently being integrated in CanoP.

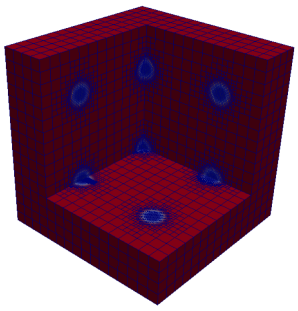
8.2 Performance tests using the two-phase application of CanoP

The first results using CanoPcode, which are presented here, are concerned with the code performances on parallel architectures. We are indeed analyzing the overheads due to the call for MPI functions and to the use of AMR comparing with uniform grids. A coarse profiling for time spent in some parts of the code is also presented, although a finer analysis should be done using dedicated tools. However, since CanoPis still under development, and as no work on optimizing it has been conducted yet, we are only interested in the main sources of computation cost here.

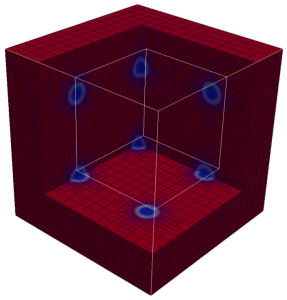
The aspects we are looking and comparing here are the number of MPI processes, the computation time (or Wall Clock Time) of the main subparts of the code, and the compression rate of the grid (whose definition is given in the introduction of Part III. A discussion on the compression errors, and on the point of using AMR while preserving the same solution's accuracy is carried out in Chapter 9: we will study there the influence of the refinement criterion on the compression rate and on the compression error. In the present chapter, however, we are not interested in the accuracy of the numerical solution, but only in the code performances according to the grid characteristics.



(a) Advection profile on one tree and a uniform mesh.



(b) Advection profiles (8 patterns) on one tree and adapted mesh.



(c) Advection profiles (8 patterns) on 8 trees and adapted mesh.

Figure 8.7: *Pictures of the initial conditions for the numerical tests, that were used to measure CanoPperformances in 3D.*

8.2.1 Presentation of the test case and computational environment

For the following scalability measurements, we choose a 3D configuration of an advection test case. The 3-equation equilibrium (Chapter 1) system is solved using the relaxation scheme with Bouchut's correction (Chapter 4), dimensional splitting, on hexahedral cells and with the following initial conditions, defined at any point M in a piece of the domain Ω around a central point O:

$$\mathbf{u} = \begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, \quad p = 1.0{\cdot}10^5\,, \quad \alpha = \begin{cases} 1.0 & \text{if} \quad d(O,M) < 0.05 \\ 0.5\left(1 + \cos\left(\pi \dfrac{d(O,M) - 0.05}{0.1}\right)\right) & \text{if} \quad 0.05 \leqslant d(O,M) < 0.15 \\ 0.0 & \text{else} \end{cases}$$

This initial condition is illustrated in Figure 8.7a. Using an idea of Burstedde, Calhoun, Mandli, and Terrel (2013), this pattern centered around O will be replicated for the weak scaling tests (as illustrated in Figure 8.7b and Figure 8.7c), so as to keep a constant load per MPI process. Indeed, it is not easy to control the number of cells with AMR grids, and especially with the heuristic refinement criteria that are used in CanoP: here, the refinement criterion is based on the variations of α, $\xi_{\text{refine}} = 1.0 \cdot 10^{-5}$ and $\xi_{\text{coarsen}} = 2.0 \cdot 10^{-6}$.

The following results were obtained on GENCI's OCCIGEN supercomputer, whose technical features can be find in CINES (2017). The number of MPI tasks per node is less or equal than 24: for N_{MPI} number of MPI processes, the number of nodes is given by $\left\lfloor \dfrac{N_{\text{MPI}}}{24} \right\rfloor$. Both p4estlibrary and CanoPwere compiled with the option flag `-O3`.

8.2.2 Strong scaling results

We are first interested in the strong scaling of CanoPfor different types of meshes. We choose the initial condition as described above, defined on a unique cubic tree, where $O = \begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix}$. We then simulate the advection of the initial profile at velocity $\mathbf{u}$ during a few milliseconds ($t_f = 2.5 \cdot 10^{-3}$), over four different grids:

- a uniform grid of level of refinement of 9, meaning that it is composed of $2^9 \times 2^9 \times 2^9$ cells;
- an adapted grid of maximum level of refinement of 9 and minimum level of 8, whose mean compression rate during the simulation is of 86 %;
- an adapted grid of maximum level of refinement of 9 and minimum level of 7, whose mean compression rate during the simulation is of 97 %;
- an adapted grid of maximum level of refinement of 9 and minimum level of 6, whose mean compression rate during the simulation is of 98 %.

The same test is performed over $N_{\text{MPI}} = 2^N$ MPI processes, where $N \in \{0, ..., 14\}$, except for the simulation over the uniform grid where $N \in \{5, ..., 14\}$.

Results of the mean time of computation per time iteration are shown in Figure 8.8a, and compared with an ideal scaling, which is defined by:

$$\tau_{\text{ideal}}(N_{\text{MPI}}) = \frac{\tau_{\text{clock}}(N_0) N_0}{N_{\text{MPI}}}.$$

Note that $N_0 = 32$ for the uniform (due to the memory cost of the computation, which prevented it from being made on a unique node), $N_0 = 1$ for the adapted grids. The parallel efficiency, defined by:

$$\eta(N_{\text{MPI}}) = \frac{\tau_{\text{ideal}}(N_{\text{MPI}})}{\tau_{\text{clock}}(N_{\text{MPI}})}$$

is presented in Figure 8.8b. Note that the reference number of processes and time of computation is not the same for the uniform or the adapted grids. Results comparing the efficiencies in Figure 8.8b should be shifted.

From these figures, one can see that the computation on the uniform grid has a correct scaling up to $1.6 \cdot 10^4$ MPI processes: its efficiency is above 0.5. For the adapted grid with one level of refinement (with $r_{\text{comp}} = 86$ %), the scaling is correct up to 256 processes. For the adapted grids with two and three levels of refinement (with $r_{\text{comp}} = 97$ % and $r_{\text{comp}} = 98$ % respectively), the scaling is correct up to 64 processes.

One can observe that for the adapted grids, there exists a critical number of processes N_{crit} above which the cost of the computation increases with the number of processes. $N_{\text{crit}} \approx 4096$ for the 1-level, 2-level and 3-level compressed grids. Above N_{crit}, the computation time is also larger on the adapted grids than on the uniform grid.

In Figure 8.8c, we compare the ratio of the computation times between the adapted and the uniform grids, given as a function of the compression rate, for different numbers of MPI processes. This figure shows that the same order of computation time can be saved by using adapted grids instead of a uniform grid up to 1024 processes, whatever the compression rate. Above 4096 processes, one can clearly see that the ratio of times is close to 1, or even above. The time savings are not perfect: the shift between the effective ratio and the ideal one is probably due to the cost of the adapting procedures.

CanoP strong scaling for the two-phase application is not perfect, especially when using adapted grids. However, time is effectively saved when using the adapted grids compared to the computation time for a uniform gird. For the case we have tested (the 3D advection test with a maximum level of refinement which is 9), this gain of computation time is true up to $N_{\text{crit}} = 4096$ processes, and it is satisfying up to 1024 processes. For N_{crit}, the number of octants per MPI process is around $3.2 \cdot 10^4$ for the uniform grid, $4.6 \cdot 10^3$ for the 1-level adapted grid and near $1.0 \cdot 10^3$ for the 2-level and 3-level adapted grids.

This number is equivalent to a subdomains of uniform grids made of $10 \times 10 \times 10$ cells. We expect that the communications for this domain decomposition are very expensive comparing with the other computation tasks.

Let us finally underline that the present results are strongly related with the two-phase application. For the spray application for example, the source term integration is the prevailing source of computational costs, and the strong scaling results are quite different (see the results for the spray solver in Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016)).

8.2.3 Time repartition between subparts of the code

Now let us look in more details at the parallel efficiencies of two subparts of CanoP: the task related to the updating of the solver (see Figure 8.2) and the task related to the adaptation of the mesh (in Figure 8.4). The adaptation task will then be decomposed between the three sub-tasks that are the refining, coarsening and 2:1 balancing tasks. We perform the same test in the same configurations as in the previous section, for a strong scaling study. The results are presented in Figure 8.9a and Figure 8.9b.

We can see in Figure 8.9a that the computation cost of the mesh adaptation functions is lower than the cost of the solver update, up to 2048 MPI processes. Then, the adaptation procedures are more expensive.

Among the tasks related to adaptation, the 2:1 balance tasks is the most expensive. Then, the refinement task necessitates more computational time than the coarsening task. This may be explained by the numerical scheme, the refinement criterion and the test case. Indeed, due to numerical diffusion, the values of α variable spread over space and the grid needs more refinement than coarsening during the computation.

Above $N_{\text{crit}} = 4096$ MPI processes, the adaptation tasks become increasingly expensive, and the strong scaling is lost. This critical number of processes corresponds to the critical number for the 2:1 and refinement tasks. The coarsening task already shows poorer results around 1024 processes.

The adaptation tasks necessitates more MPI communications than the solver task for the two-phase application, and is thus critical for a too large number of MPI processes. Once more, let us note that this observation is strongly related to the numerical scheme and other parameters of the two-phase application. In the future, for this special test case, we could conduct a study on the frequency at which the mesh adaptation procedure is applied. Indeed, in the present case, mesh adaptation occurred at each time iteration. However, changing this procedure may have an influence on the compression error (which is studied in Chapter 9), since we use heuristic refinement criteria. It should be interesting also to design better criteria or consider a multiresolution approach that ensures to keep a constant compression error.

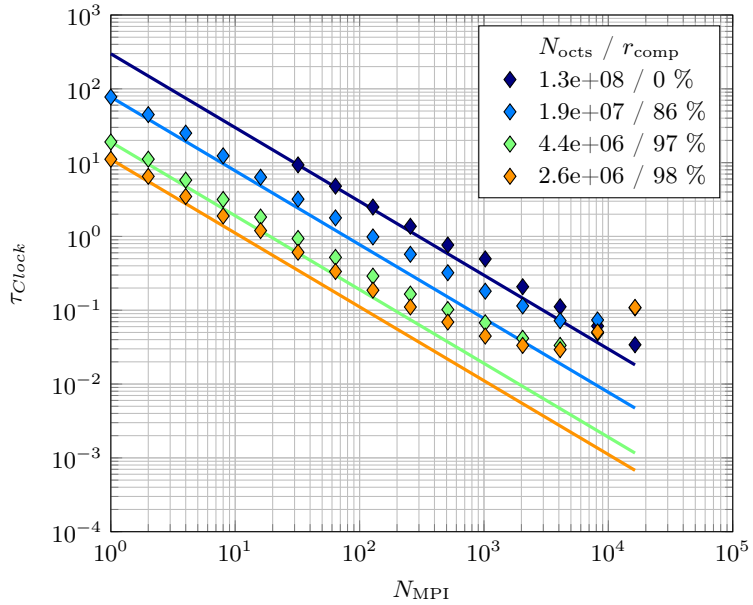
8.2.4 Weak scaling results

An other characteristics of a parallel code is its weak scaling, showing how the parallel overheads evolve for a same amount of work per MPI process.

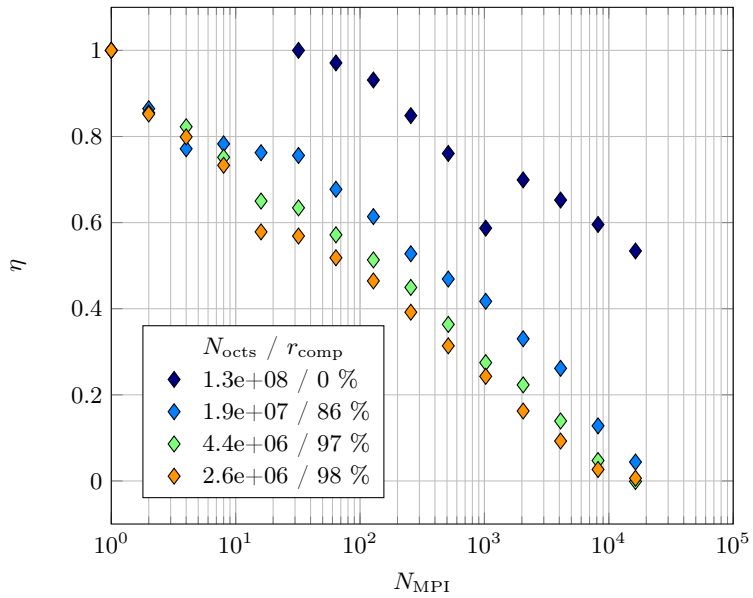
The weak scaling measures have been performed on three mesh configurations:

- a uniform grid over a unit cube, whose level of refinement is 6 for 2 MPI processes ($2^6 \times 2^6 \times 2^6$ total octants), level 7 for 16 MPI processes...
- a forest of unit cubes, with a maximum level of refinement of 6 and a minimum of 2 in each tree. The pattern defining the initial condition is reproduced once for 2 MPI processes, 8 times for 16 MPI processes...
- a unit cube containing $2N_{\text{MPI}}$ times the initial condition and whose maximum and minimum levels of refinement depend on N_{MPI}.

Using these configurations, the resulting number of octants per MPI process is given in Figure 8.10a, and the wall-clock times per iteration of the simulation are presented in Figure 8.10b. Figure 8.10c shows the

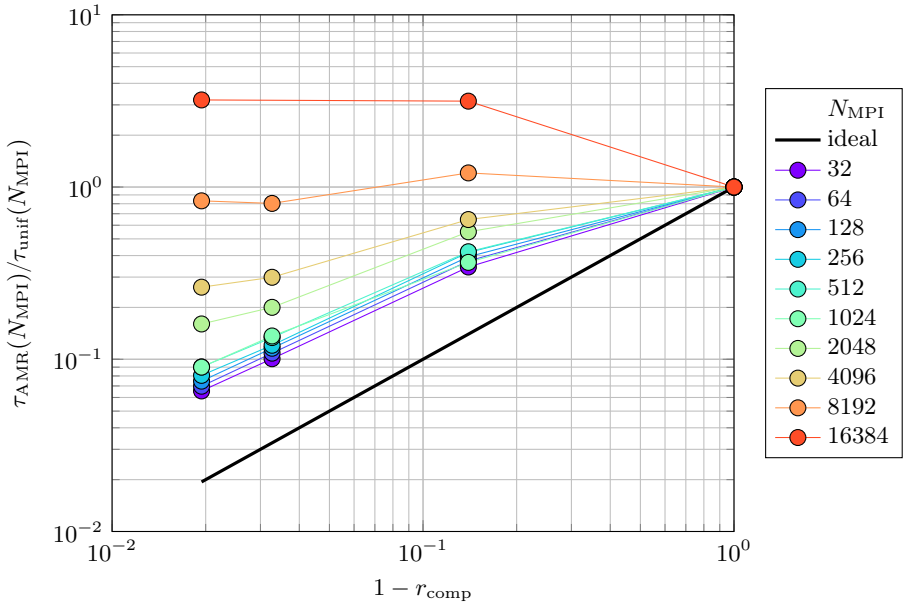


(a) Number of MPI process N_{MPI} vs wall clock time τ_{Clock} for different compression rates r_{comp}. Thick lines represent the ideal scaling.



(b) Number of MPI process N_{MPI} vs parallel efficiency $\eta = \tau_{Clock}/(N_{\text{MPI}}\,\tau_1)$ for different compression rates r_{comp}.

Figure 8.8: *Strong scaling results for a 3D simulation case.*



(c) Compression rate r_{comp} vs relative computation time between adapted and uniform grids for different numbers of MPI processus.

Figure 8.8: *(continued) Strong scaling results for a 3D simulation case.*

relative time for each test with respect to the computation time for the configuration of 2 MPI processes.

For both Figure 8.10b and Figure 8.10c, a perfect scaling would be a constant whatever the number of processes. We can note however that the wall-clock time increases with the number of MPI processes. When comparing the times spent in the solver and in the mesh adaptation parts, we can see that, whether the solver part is more expensive, it has a better weak scaling than the mesh adaptation part.

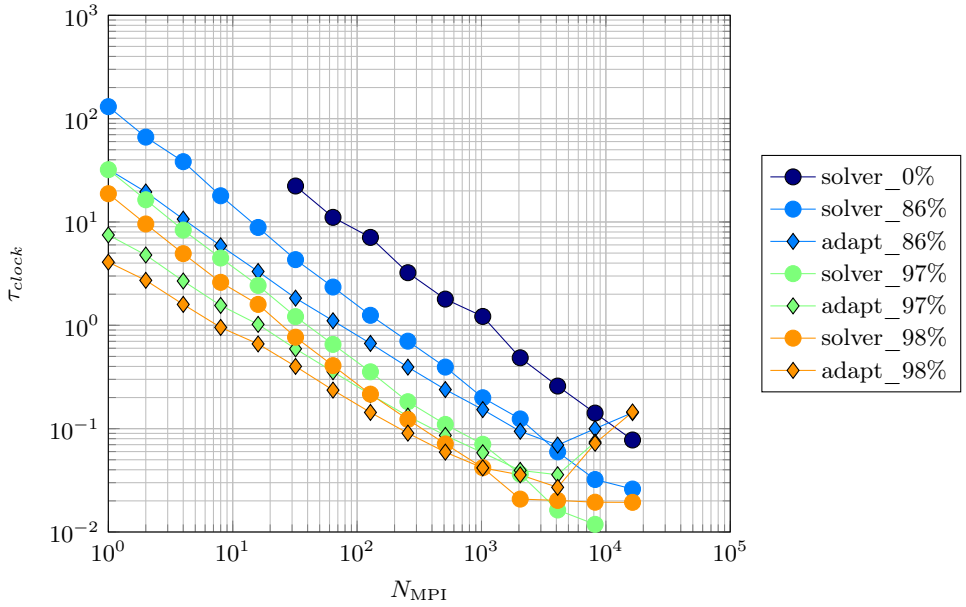
8.2.5 Comparisons with standard resolution on uniform grid

Here we compare the execution times of CanoP and a simple 2D code, written in C language and that we call toyCode. This code is dedicated to the resolution of the 3-equation system on cartesian grids with Relax-B scheme presented in Chapter 4. It is not parallel and no specific optimization work has been done on it, except using the compilation option flag `-O3`.

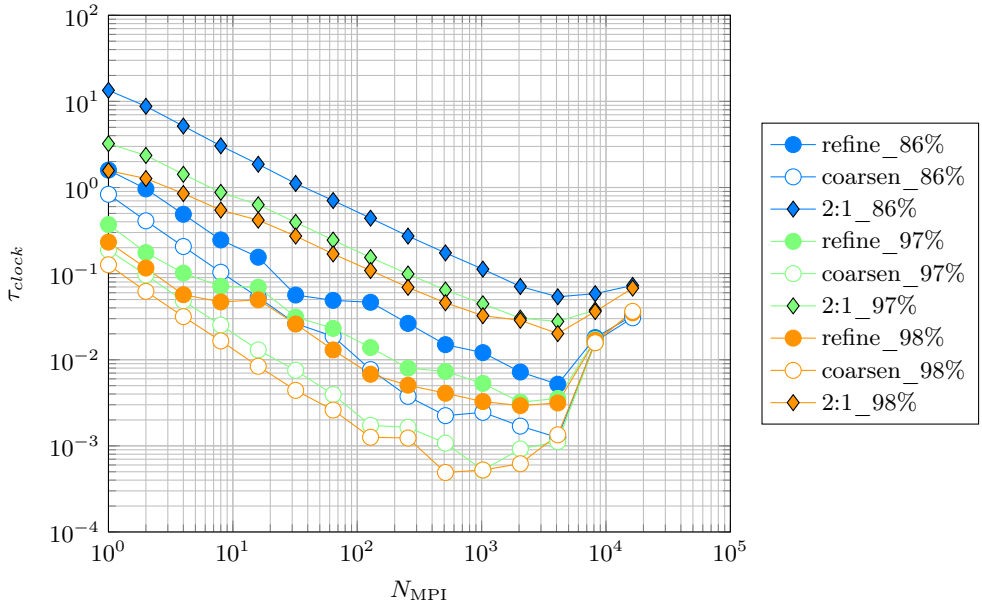
For a same advection test in 2D, we compare then the computation times of CanoP and toyCode with equivalent grids (uniform or adapted with a same equivalent uniform grid), of 1024×1024 cells, for 328 time iterations. Results on one core are presented in Figure 8.11: the execution time for toyCodeis 123 s and the time for CanoP on the same uniform grid is 284s. Then, for the adapted grids, the computation time is given as a function of the compression rate r_{comp} (more precisely in function of $1 - r_{comp}$). The point is to determine the minimum compression rate that enables a faster computation with CanoP. The compression rate, for which the two execution times are of the same order, is around 81 %: this corresponds to two levels of compression. Let us point out here that whatever the compression rate, the accuracy of the solution is kept constant.

Over uniform meshes, comparisons between CanoP and toyCode show a ratio of their execution times that fluctuates between 2.2 and 4. However, this comparison is to be qualified due to several differences between the two codes:

- the scheme implementations differ slightly between the two codes,
- CanoPcontains many more features and options than toyCode. Its structure is consequently more complex and involves more operations (for instance, data for solving the Riemann's problems are reconstructed, whether the scheme is second order or not). In that point of view, some optimizing

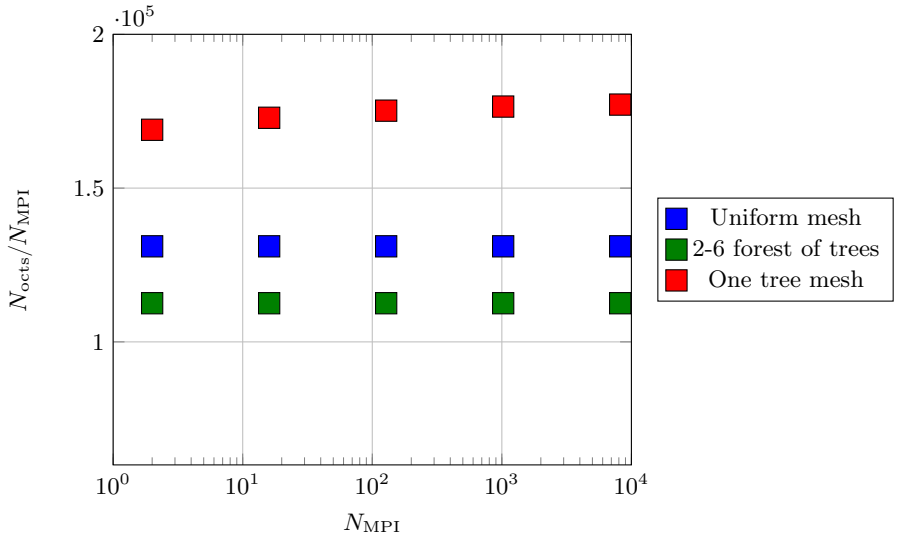


(a) Wall clock times for solver and mesh adaptation parts.

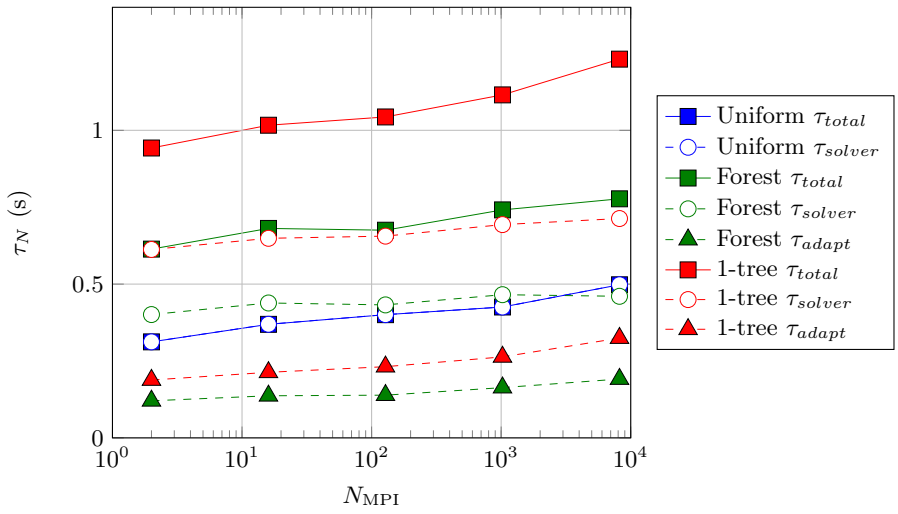


(b) Wall clock time for refining, coarsening and 2:1 balancing parts.

Figure 8.9: *Strong scaling results for a 3D simulation case.*

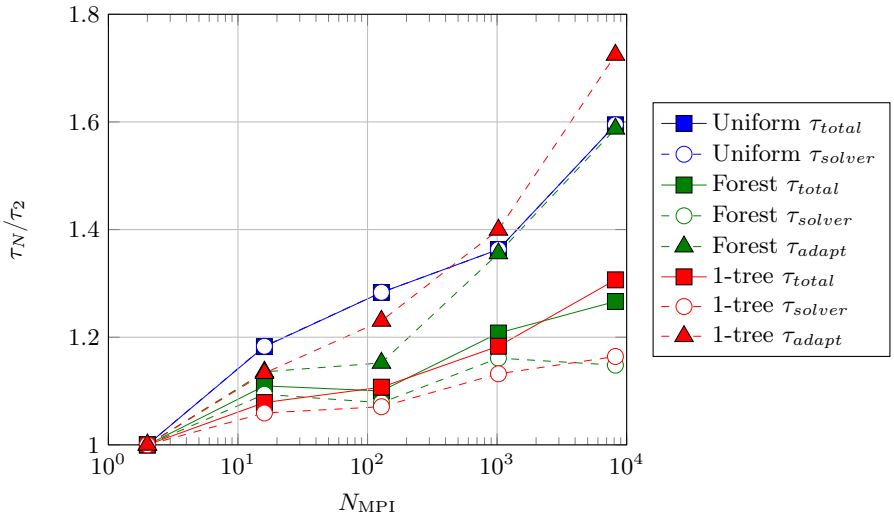


(a) Load per MPI process for the different computations.



(b) Computation time for the tests involving different numbers of MPI processes.

Figure 8.10: *Weak scaling test. Computation time and load repartition.*



(c) Relative computation time for the tests involving different numbers of MPI processes.

Figure 8.10: *(continued) Weak scaling test. Computation time and load repartition.*

could be considered in the future.

- CanoPis a parallel code, and as shown in the previous paragraphs, has an acceptable scaling up to some thousands of cores. On the other hand, it would be easy to parallelize toyCodeusing classic domain decomposition.

Moreover, this kind of comparison depends strongly on the model, the scheme and on the test case. Indeed, for the spray model for example, the source term is an expensive and local task. Using AMR for this model is clearly advantageous, as shown in Essadki, de Chaisemartin, Massot, Laurent, Larat, and Jay (2016). The 4-equation and 5-equation two-phase models also involve the integration of source terms (see Chapter 5), and thus may be more efficiently solved with AMR.

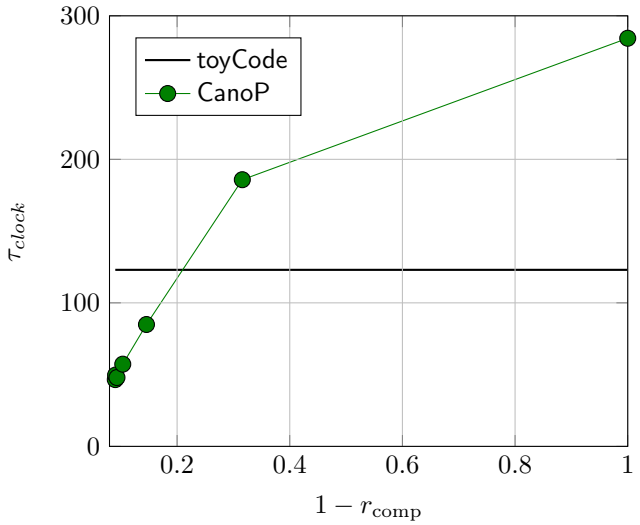


Figure 8.11: *Execution time for a 2D advection test with respect to the compression rate (for an equivalent grid of size* 1024×1024*).*

8.3 Improvements of CanoP code and works in progress

Many improvements of CanoP are currently under progress. Today, the two main axes are:

- an integration of higher-order methods, that necessitate to change the heavy data structures of CanoP,
- an externalization of the memory management for these data, so as to better use cache locality and have a better flexibility on the size of the data.

These two aspects are detailed below.

8.3.1 Subgrids and degrees of freedom

Triangular grids

One first and simple modification of the sub-cells degrees of freedom is to divide each quadrangular cell into to triangular subcells. From a quadrangular mesh, and particularly from the cell-based mesh of p4est, we propose here a very simple operation that enables to get a triangular mesh, and to perform tests of Low-Mach number flows (see details about this topic in Chapter 6). This operation is presented below for the 2D case and propositions are made for their adaptation to 3D (since it has not yet been tested). It relies on a modification of the user's data structure, so as to take into account two inner degrees of freedom in each cell.

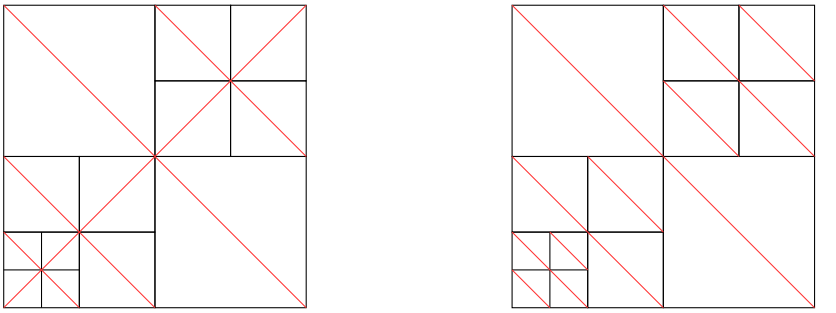
With the same idea as in p4est of identifying cells using indices only, we are looking for a simple management of the triangular grid. Two configurations are considered and illustrated in Figure 8.12: with the first configuration, quadrangular cells are divided into 2 triangles, with an orientation of their hypotenuse according to the Morton's index of the cells, while in the second configuration, all triangles are oriented the same way. For sake of simplicity in the 2D case, we have selected the second option. We will comment this choice regarding the 3D case at the end of this paragraph.

We then give a number to each triangle, in agreement with the face numbering operated by p4est, as shown in Figure 8.13a: triangle 0 if it contains the face 0, triangle 1 otherwise. Going through all the triangular cells of the grid will then be made following first the quadrants indices, and secondly the triangles indices. We can note that, in a tree and through any face of a given triangle, the neighboring triangle's number is the opposite one.

Refining and coarsening triangles is done in the same way as for quadrangles (see illustration in Figure

8.13b): when the quadrant is divided into four new quadrants, the triangles it contains are also split into four new triangles. This comes down to keep two triangles per quadrangular cell.

Dimensional splitting is not relevant with triangular grids, so we use an unsplit update of the transport part of the equations with Relax-B. At the faces of the cells, fluxes are computed in the referential of the normal to the face (which is easily computed for the diagonal face), and then projected in the global referential.



(a) Division of each quadrangular cell into two triangles oriented in the same direction.

(b) Division of each quadrangular cell into two triangles oriented in two different directions.

Figure 8.12: *Changing an AMR quadrangular mesh into a triangular mesh.*

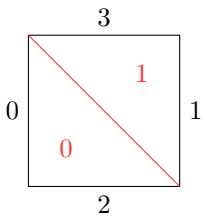
In 3D, there are again several options leading to the division of hexahedral cells into tetrahedral ones. The least number of such created tetrahedra is 5, but the simplest division leads to 6 tetrahedra, and is used in the tetrahedral version of p4est in Burstedde and Holke (2016). This is illustrated in Figure 8.14. This 3D version has not yet been implemented in CanoP, but is does not present any difficulty.

Other applications based on multiple degrees of freedom per cell

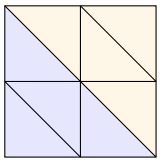
Triangular grids, built from the quadrangular version of p4est can be seen as a first step towards the definition of multiple degrees of freedom inside each leaf of the tree. This approach is currently under study for the implementation of MOOD (Clain, Diot, and Loubère (2011)) or Discontinuous Galerkin (DG) (see a recent review of this method in Shu (2013)) high-order schemes.

For MOOD schemes, the increase in scheme order is based on a polynomial reconstruction of the solution inside each cell, using a large stencil of neighbors. This stencil increases with the order of the method. As it is difficult and costly to find neighbors of neighbors in p4est, the idea is then to add subgrids (or patches) in each p4est cell. This way, all required neighbors are provided with only one call of neighboring cells in p4est. This method is being implemented for the hydrostatic Euler application in CanoP.

In DG approach, degrees of freedom of a finite element method are used to set the nodal values of the polynomials basis. This method has been mentioned for the fractional-moment model in Essadki, Jung,

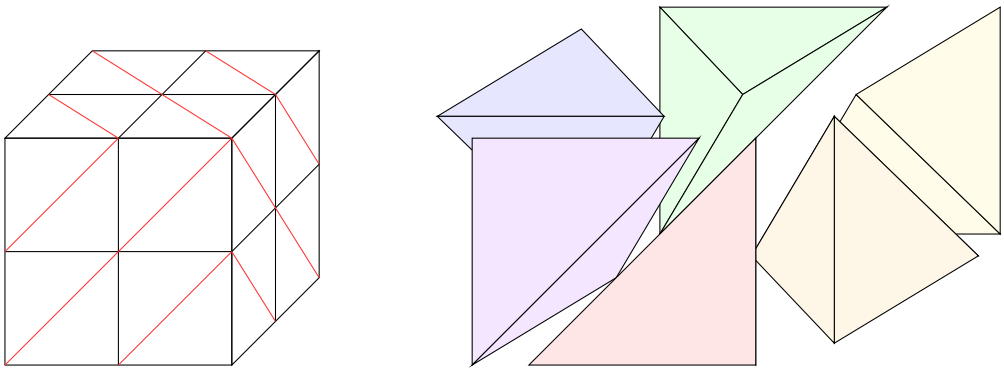


(a) Face indexing according to p4est and triangle numbering.



(b) Coarsening of triangular cells.

Figure 8.13: *Changing an AMR quadrangular mesh into a triangular mesh.*



(a) Tetrahedral 3D grid. **(b)** Division of a hexahedral cell into 6 tetrahedra.

Figure 8.14: *Changing a hexahedral mesh into a tetrahedral mesh.*

Larat, Pelletier, and Perrier (2017), and is currently being implemented in CanoP.

8.3.2 Memory externalization

The need for uncoupling the user's data allocation from the management done by p4est has raised from the observations that, although p4est quadrants data are stored using the Morton index, it is not the case of the user's data pointed by these quadrants. Indeed, heavy data are allocated using a memory pool principle by the SC library. This mismatch in the cache memory can lead to overheads, which increase with throughout simulation run, when the grid is strongly modified.

To tackle this issue, we have decided to let the allocation of these data be performed by CanoP. This new memory management will moreover be useful when coupling different CanoP's solvers, for which the size of the data may differ. Finally, it may be useful for a *hp* refinement strategy with the DG method. This means that both spatial discretization and the order of the polynomial are changed according to the need of accuracy of the solution.

8.3.3 Other

Other improvements of CanoP consist in the implementation of Poisson's solver, for the hydrostatic Euler equations, or also for instance, to solve incompressible Navier-Stokes.

Finally, a cleaning of some functions and an optimization of some parts of CanoP would help improve the code performances.

8.4 Conclusion

CanoP code is an applicative code of the tree-based AMR framework proposed by the p4est library. CanoP has been developed so as to allow the integration of different finite volume applications (models and numerical schemes) and the use of a set of numerical features, related to the finite volume methods. CanoP code is still under development: it is modified to be more generic through the object-oriented framework, more performant by means of an externalization of the memory management, new high-order methods are implemented and tested and a coupling between the applications is envisioned.

The two-phase application was the second application of CanoP. It solves the 3-equation model with Relax-S, Relax-Band HLLC scheme and may be used at the second order in space and time with the MUSCL-Hancock scheme. The verifications of the accuracy of these methods in an AMR context is presented in Chapter 9. Here, we have tested the performance of CanoP for this application and in a HPC architecture. The scalability results are satisfying, knowing that no optimization work has yet been

performed for CanoP. A comparison with a simple 2D code shows a factor of 2 to 4 between the two codes execution times. Once more, these are encouraging results, which could probably be improved by some code optimization.

Among the other development opportunities, one can note that tree-based grids are at the heart of the wavelet-based multi-resolution approaches of Harten (1994). However, their use is somehow different than in AMR approaches. In particular, the prediction stage in the multi-resolution approach involves stencils of neighboring cells, which are larger than the stencil required for the evaluation of the heuristic criteria in AMR and which may be expensive to obtain, due to recursive research of neighbors. We can wonder in what extent these methods can be integrated in CanoP.

Part IV

AMR simulations of separated-phase flows configurations

Abstract

In this part, we present a series of numerical tests of 1D, 2D and 3D AMR configurations. The results have been obtained with the two-phase application of CanoP code. In Chapter 9, we study the effects of the refinement criterion on the compression error for an advection test. Then, we perform two standard Riemann problems. The low-Mach problem is introduced with a 2D test of a rising bubble. Finally, a 2D dam break test with an obstacle, and a simple 3D dam break are presented in Chapter 9. Other configurations of dam breaks are studied in Chapter 10. There, we compare the results of the compressible two-phase model with experimental data and an incompressible model. Another study of the low-Mach issue is also proposed for the first moments of the dam break problem.

Chapter 9

AMR verification tests

9.1 Presentation of the chapter

In this first chapter dedicated to numerical results, we are presenting a series of numerical tests that have been designed to experience the numerical schemes, presented in Chapter 4 and Chapter 6, and their integration in the CanoP code. We shall focus on numerical issues that are specific to the AMR context:

- the effects of the non-conforming character of the mesh, that could deteriorate the numerical solution at the interface between large and smaller cells,
- the effects of dynamically changing grids, that could introduce numerical errors during refinement or coarsening,
- the influence of the choices of both the refinement criterion and the refinement thresholds on the numerical solution.

In order to examine these issues, we shall test our discretization strategies and implementation against classic Riemann problems and multi-dimensional flows.

Another aspect that is emphasized with adapted grids is the difficulty related to the low-Mach number flows, as we will see with the test of the rising bubble. This test will then be used to test the all-regime scheme that was presented in Chapter 6, as well as the triangular grids (Chapter 8).

The last two tests deal with the simulation of dam breaks in both 2D and 3D, and in complex domains (more complex than unit square domains). A further study will be presented in Chapter 10 including comparisons with experimental data.

In this chapter, we are solving the 3-equation system (derived in Chapter 1) which reads:

$$\begin{cases} \partial_t \rho & + & \mathrm{div}_{\mathbf{x}}\left(\rho \mathbf{u}\right) & = & 0 \\ \partial_t (\rho Y) & + & \mathrm{div}_{\mathbf{x}}(\rho Y \mathbf{u}) & = & 0 \\ \partial_t \left(\rho \mathbf{u}\right) & + & \mathrm{div}_{\mathbf{x}}\left(\rho \mathbf{u}\mathbf{u}^T + p\mathbf{Id}\right) & = & \rho \mathbf{g} \\ p = p_1\left(\frac{\rho Y}{\alpha(\rho,Y)}\right) = p_2\left(\frac{\rho(1-Y)}{1-\alpha(\rho,Y)}\right) & & & & \end{cases} \tag{9.1}$$

where $\mathbf{g}$ stands for the gravity source term ($\mathbf{g} = \mathbf{0}$ in the first tests), and the pressures p_k, $k = 1, 2$ are computed with linearized barotropic pressure laws:

$$p_k = p^o + c_k^2(\rho_k - \rho_k^o),$$
$$\text{with} \quad p^o = 1.0 \cdot 10^5, \quad c_1 = 3.4 \cdot 10^2, \quad c_2 = 1.5 \cdot 10^3,$$
$$\rho_1^o = 1.0, \quad \rho_2^o = 1.0 \cdot 10^3, \quad \rho_1 = \frac{\rho Y}{\alpha(\rho, Y)}, \quad \rho_2 = \frac{\rho(1-Y)}{1-\alpha(\rho, Y)}.$$

Let us also define the notion of **compression level**, that will be used in the discussions. The level of compression refers to the number of refinement levels separating the finest levels from the coarsest levels

of refinement. For instance, a 2D grid whose finest level is 12 (finest cells are $2^{2 \cdot 12}$ times smaller than the whole domain) and coarsest level is 3, has a level of compression of $L_{\text{comp}} = 12 - 3 = 9$. In the following, this quantity is noted L_{comp}

9.2 A 2D advection test case and influence of the refinement criterion

The first test is a simple advection of a volume fraction profile. This test will provide a ground for convergence studies and for a discussion on the refinement criterion. This test possesses a simple analytical solution, that allows convergence studies and a discussion on the choice of the refinement criterion, since L_1- and L_2-errors are easy to estimate.

9.2.1 Presentation of the test

Initial and boundary conditions: The initial condition is given by:

$$\begin{cases} p(0,\mathbf{x}) &= p_0, \\ \alpha(0,\mathbf{x}) &= f(\mathbf{x}), \\ \mathbf{u}(0,\mathbf{x}) &= \mathbf{u}_0, \end{cases} \tag{9.2}$$

with $p_0 = 1.0 \cdot 10^5$, $\mathbf{u}_0 = (1,1)^T$ and:

$$f(\mathbf{x}) = \begin{cases} 0 & \text{if} \quad |\mathbf{x}-\mathbf{x}_0| > L_0, \\ 0.5\left[1+\cos(\pi\frac{|\mathbf{x}-\mathbf{x}_0|-l_0}{L_0-l_0})\right] & \text{else if} \quad |\mathbf{x}-\mathbf{x}_0| > l_0, \\ 1 & \text{else.} \end{cases}$$

We choose, in a unit square domain, the following values for l_0, L_0 and $\mathbf{x}_0$:

$$\begin{aligned} l_0 &= 0.05, \\ L_0 &= 0.15, \\ \mathbf{x}_0 &= (0.5, 0.5)^T. \end{aligned}$$

An illustration of the initial condition is given in Figure 9.1.

The domain is assumed to be periodic. Then the analytical solution is given at time $t > 0$ by:

$$\begin{cases} p(t,\mathbf{x}) &= p_0, \\ \alpha(t,\mathbf{x}) &= f(\mathbf{X}(t,\mathbf{x})), \\ \mathbf{u}(t,\mathbf{x}) &= \mathbf{u}_0, \end{cases}$$

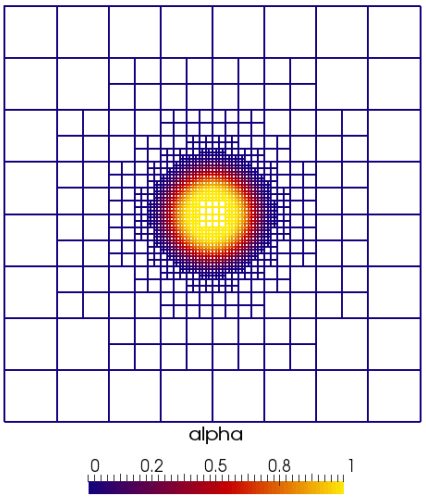
and:

$$\mathbf{X}(t,\mathbf{x}) = \begin{pmatrix} x - u_{0,x}t - \lfloor x - u_{0,x}t \rfloor \\ y - u_{0,y}t - \lfloor y - u_{0,y}t \rfloor \end{pmatrix}, \quad \mathbf{u}_0 = (u_{0,x}, u_{0,y})^T, \quad \mathbf{X} = (x,y)^T.$$

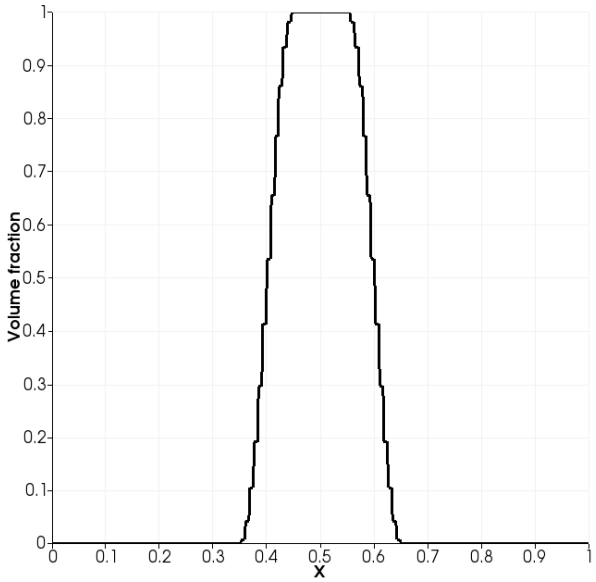
This test should preserve uniformity of pressure and velocity (see Appendix A). In practice, the final time for the following tests is $t_f = 2.5 \cdot 10^{-1}$s, and numerical errors are given by the L_1- and L_2-norms of the difference between the numerical solution and the analytical one.

Numerical scheme and refinement criterion: The numerical solutions are computed using Relax-B (see details in Chapter 4) applied to system (9.1) with $\mathbf{g} = \mathbf{0}$. We also use directional splitting, and for second order results, the MUSCL-Hancock reconstruction presented in Chapter 6. The CFL coefficient is 1 for first-order schemes and 0.5 for second-order ones.

In the following, the criterion used for AMR grids is based on the relative difference of the mixture density ρ between two cells (see the definition of this criterion in Chapter 6, section 6.4). The value of the refinement and coarsening thresholds, ξ_{refine} and ξ_{coarsen}, will be given for each test (see their definition in Chapter 6, section 6.4).



(a) A grid adapted to the profile of the density ρ (that is related to the profile of the volume fraction α).



(b) Profile of α along the $x-$axis.

Figure 9.1: *Illustration of the volume fraction profile for the advection test case at the initial condition. The smallest level of refinement is 3 ($\Delta x_{max} = 0.125$) and the highest is 7 ($\Delta x_{min} = 7.81 \cdot 10^{-3}$)*

Remark 37. *The choice of using $\Delta\rho$ rather than using the volume fraction $\Delta\alpha$ was made by noticing that the relative difference between two cells, could be arbitrarily large when α tends to 0 (which is the case with the initial conditions defined above).*

With the following tests, we study the following properties of the numerical methods that have been implemented in the CanoP code:

- stability of the directional splitting for CFL=1.0 at first order and 0.5 at second order, and preservation of the uniform velocity and pressure profiles,
- order of convergence of the relaxation scheme with Relax-B scheme,
- convergence of the AMR solution and effect of the refinement threshold.

Finally, we shall discuss the choice of the refinement criterion and thresholds and their influence on both the compression errors and compression rates.

9.2.2 Test of the first order scheme

In Figure 9.2 and Figure 9.3, we present the numerical errors (in L_1- and L_2-norm) in function of the spatial discretization and for different grids. For uniform meshes (the darkest blue curves), the abscissa stands for the side length of one cell. For *compressed* (adapted) grids, the abscissa gives the side length of the finest cells Δx_{min}, that is also the side length of one cell in an equivalent uniform grid. So we can compare solutions that have the same equivalent uniform grid. The colors of the curves of the following series of figures are associated with grids with the same compression levels L_{comp}.

In Figure 9.2, the refinement thresholds are $\xi_{\text{refine}} = 5.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 2.5 \cdot 10^{-2}$, while in Figure 9.2, these thresholds are lower: $\xi_{\text{refine}} = 1.0 \cdot 10^{-1}$ and $\xi_{\text{coarsen}} = 5.0 \cdot 10^{-3}$.

In both figures, one can first observe that, in the limit of small Δx_{min}, the order of convergence is close to 1, whatever the compression level. However, in this limit, the numerical error increases for increasing compression levels: the difference between the numerical errors for the uniform grid and the compressed grids is in fact the compression error ϵ_{comp}, as defined in Introduction, section 0.4.3. This

compression error is larger in Figure 9.2 than in Figure 9.3. In Figure 9.2, the compression error value even deteriorates the order of convergence of the solution.

One can also note that for some compressed grids, the numerical error is smaller than the numerical error of its uniform equivalent grid. This can be explained by the CFL condition that is less restrictive in the compressed grids, because zones of the solution where the sound velocity is the highest are not necessarily well refined.

The numerical errors of both uniform and compressed grids are then of same order down to a certain cell size, that we will note Δx^o_{min}. For grids whose cells are finer than Δx^o_{min}, the compression error prevails over the scheme error. The value of Δx^o_{min} depends on the choice of ξ_{refine} and ξ_{coarsen}, as one can see by comparing Figure 9.2 with Figure 9.3: when one decreases ξ_{refine}, one also decreases Δx^o_{min}.

On the other hand, we present in Figure 9.4 the effective compression rates r_{comp} of the different grids (defined in the introduction of Part III, section III). The abscissa and colors of these figures are the same as for Figure 9.2 and Figure 9.3.

As for the compression errors, we can note that ξ_{refine} (and ξ_{coarsen}) have an influence on r_{comp}: r_{comp} is increasing when ξ_{refine} (and ξ_{coarsen}) increases. This means that for higher values of ξ_{refine} and ξ_{coarsen}, the grid is less sensitive to the variations of the numerical solution.

We finally compute the relations between r_{comp} and ϵ_{comp} for two different Δx_{min} (that are visually identified on Figure 9.2, Figure 9.3 and Figure 9.4 by the blue and green vertical dashed lines). In Figure 9.5, the blue curve stands for the largest values of ξ_{refine} and the green curve for the smallest values of ξ_{refine}. We study here the ratio of the error for the compressed grid ϵ_{AMR} over the error for the uniform one ϵ_{unif}.

One can see that the values of r_{comp} can not increase infinitely: they are bounded by a value $r_{\text{comp}_{\text{max}}}$ that is strictly lower than 1, and that depends on the value of ξ_{refine}. Indeed, one can see that the lower the value of ξ_{refine}, the lower $r_{\text{comp}_{\text{max}}}$. However, choosing small values of ξ_{refine} ensures a lower compression error ϵ_{comp}: the green curves are always below the blue ones, and especially in the second figure, where the ϵ_{AMR} is of the same order as ϵ_{unif}.

Let us summarize here the main observations made for the first-order numerical solutions:

1. the convergence rates of all grids are in agreement with the order of accuracy of the numerical scheme (first-order here),
2. lowering the refinement threshold helps reducing the compression errors in the limit of small cell sizes,
3. lowering the refinement threshold also limits the compression rates of the grids.

9.2.3 Test of the second order scheme

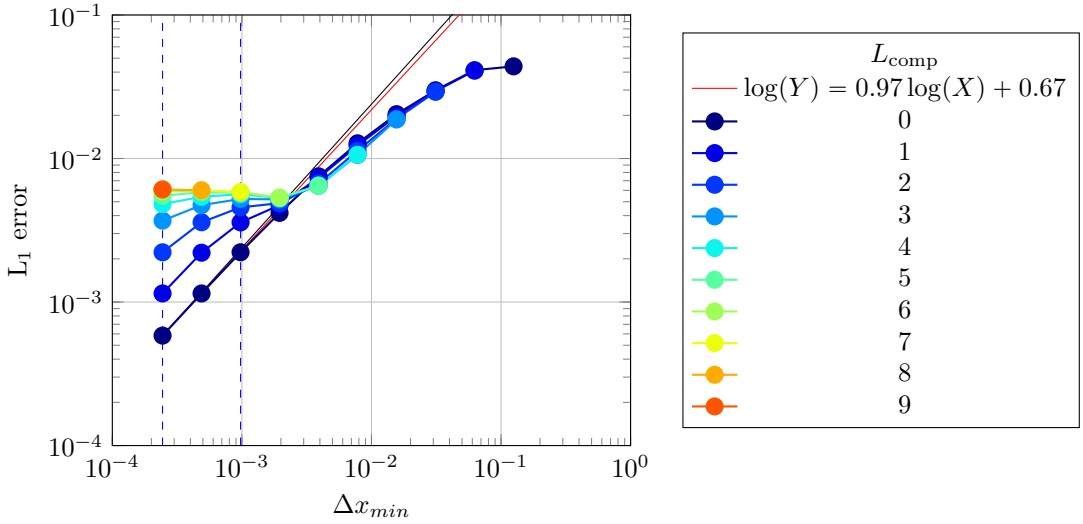
In the same way as for the first order scheme, we present here a convergence study of the second-order scheme, using the MUSCL-Hancock reconstruction. The results are presented in Figure 9.6.

The orders of convergence for uniform grids are: 1.83 in the L_1 norm, 1.68 in the L_2 norm.

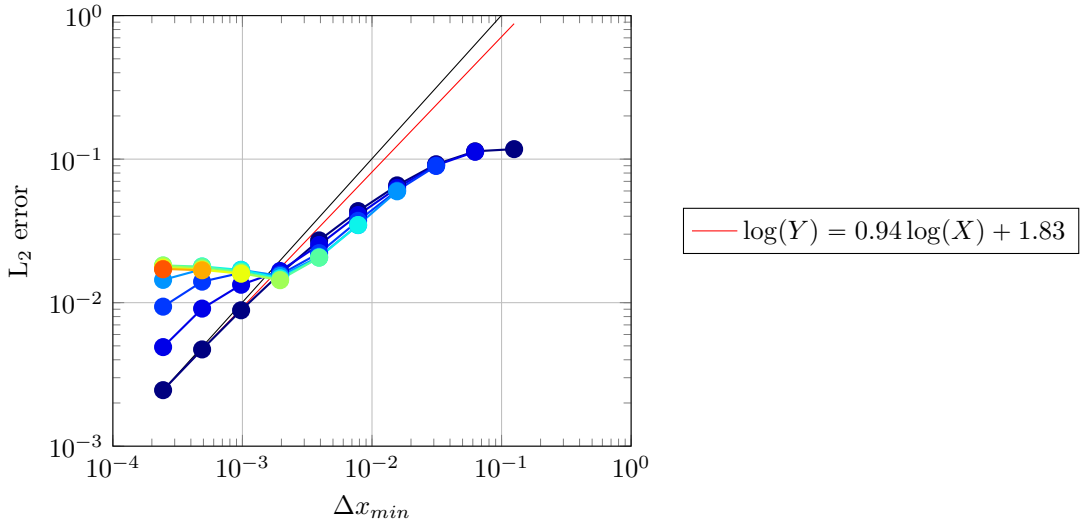
We can see again that the choice of the value for ξ_{refine} has an influence on the accuracy of the solution: for a given value of ξ_{refine}, ϵ_{comp} is negligible with respect to the whole numerical error down to a discretization level Δx^o_{min}. For grids whose finest cells are beyond Δx^o_{min}, ϵ_{comp} prevails over the other scheme truncation error. We can note that the value of Δx^o_{min} for the second-order scheme is larger than the value of Δx^o_{min} for the first-order scheme: this means that the prevailing effects of ϵ_{comp} appear on grids that are coarser than for the first-order scheme. This is due to the reduction of the truncation error for second-order schemes, while ϵ_{comp} is only related to the AMR grid.

Consequently, even for $\xi_{\text{refine}} = 5.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 2.5 \cdot 10^{-2}$, the numerical errors are not satisfying yet on the finest grids. One could also wonder whether the convergence rate of the compressed grids is not damaged.

The study of the compression rates yields similar results as the first-order case of previous section (see

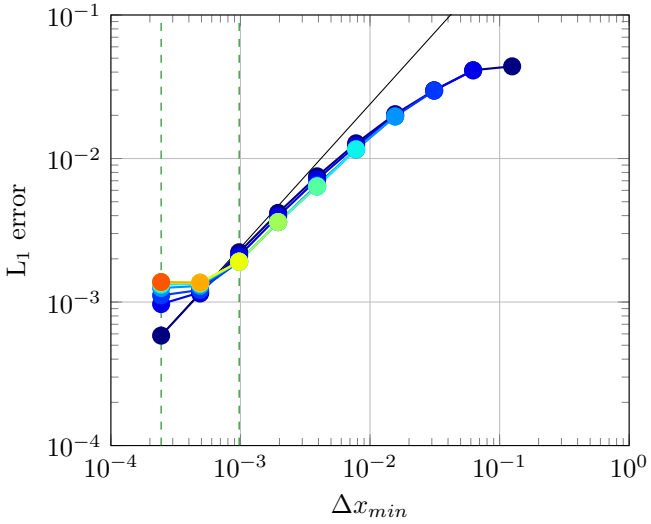


(a) L_1 error comparing with the analytical solution.

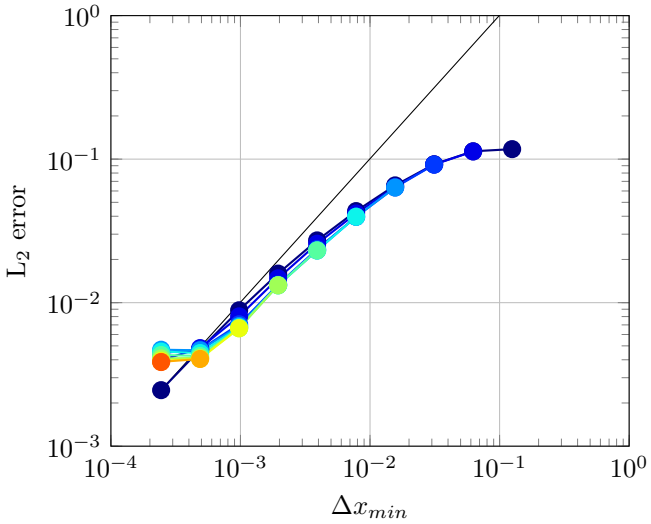


(b) L_2 error comparing with the analytical solution.

Figure 9.2: *Advection test case on different AMR grids (defined by different values of* L_{comp}*) for* $\xi_{refine} = 5.0 \cdot 10^{-2}$ *and* $\xi_{coarsen} = 2.5 \cdot 10^{-2}$*. First-order Relax-B scheme.*

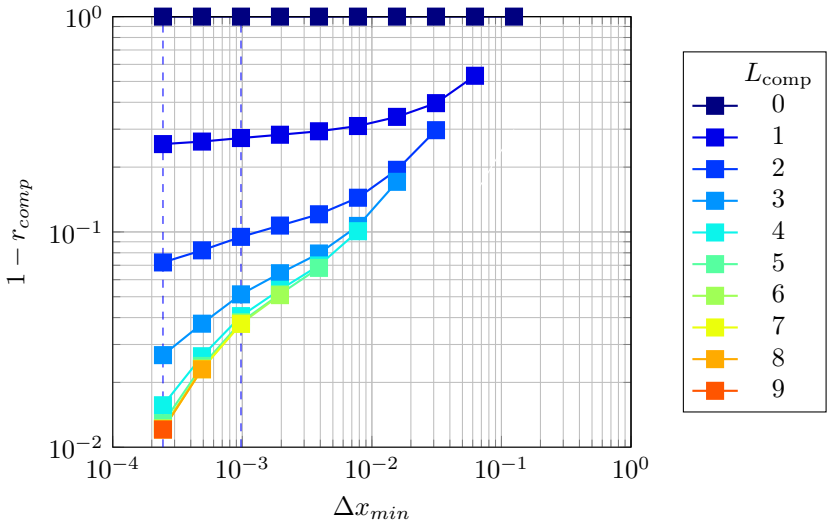


(a) L_1 error comparing with the analytical solution.

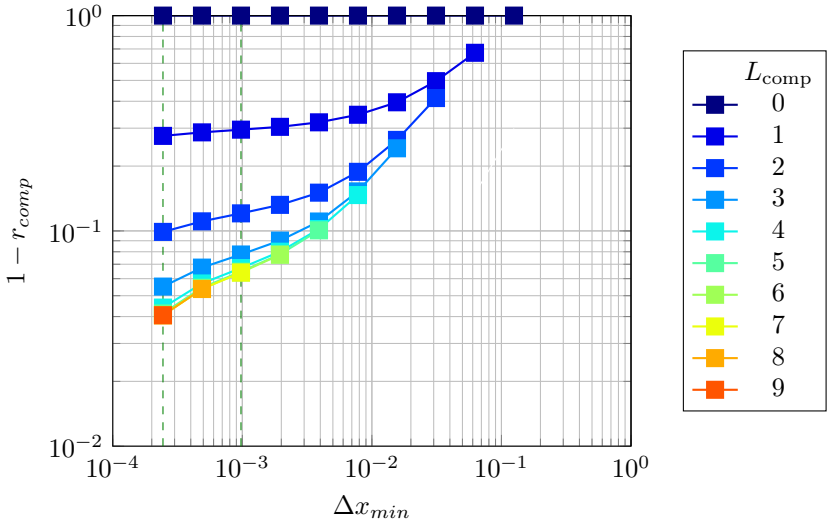


(b) L_2 error comparing with the analytical solution.

Figure 9.3: *Advection test case on different AMR grids (defined by different values of* L_{comp}*) for* $\xi_{refine} = 1.0 \cdot 10^{-2}$ *and* $\xi_{coarsen} = 5.0 \cdot 10^{-3}$*. First-order Relax-B scheme.*

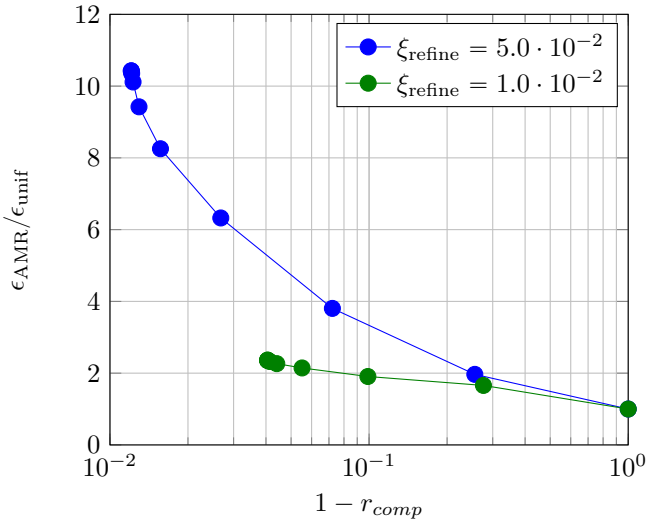


(a) Effective r_{comp} at the end of the simulation for $\xi_{\text{refine}} = 5.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 2.5 \cdot 10^{-2}$.

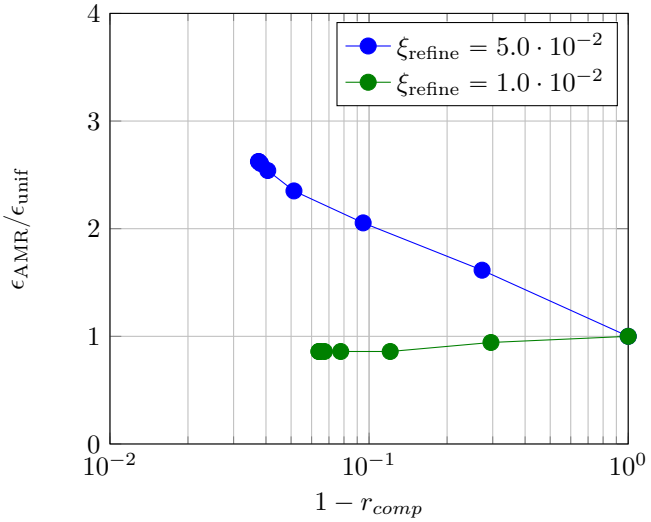


(b) Effective r_{comp} at the end of the simulation for $\xi_{\text{refine}} = 1.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 5.0 \cdot 10^{-3}$.

Figure 9.4: *Advection test case on different AMR grids (defined by different values of L_{comp}) and different values of ξ_{refine} and $\xi_{coarsen}$.*

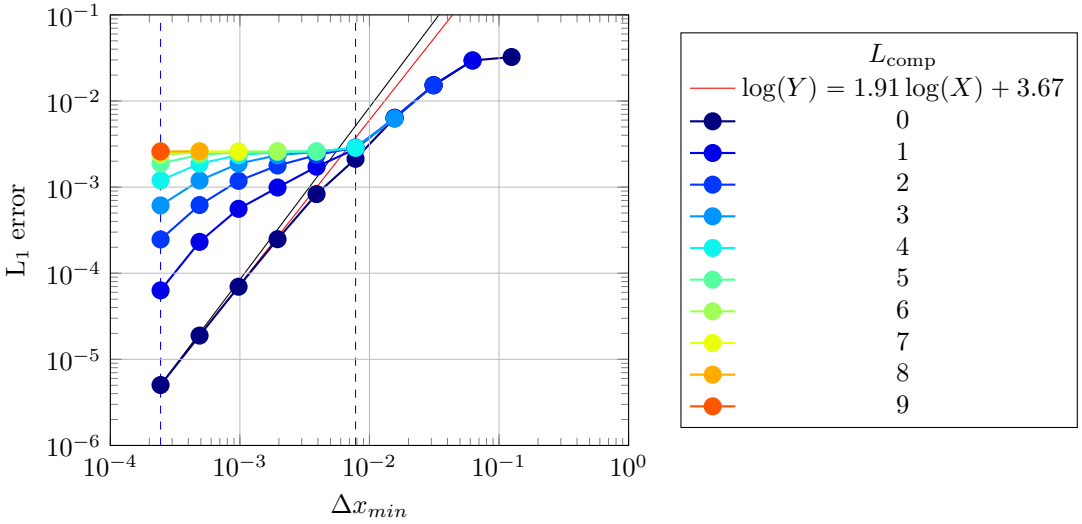


(a) Equivalent uniform grid made of $2^{12} \times 2^{12}$ cells.

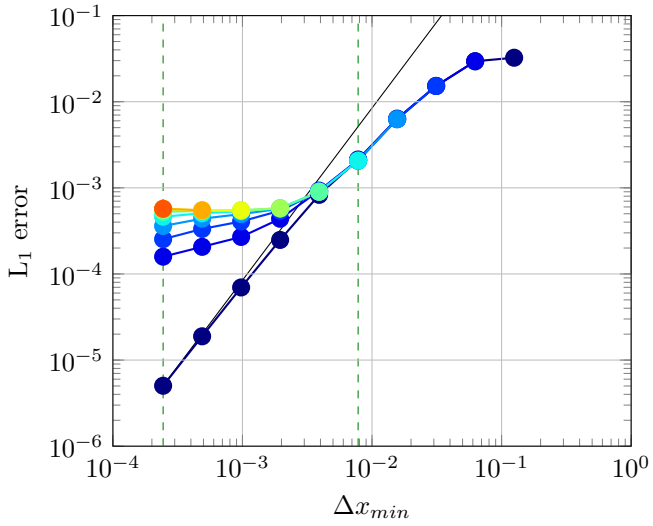


(b) Equivalent uniform grid made of $2^{10} \times 2^{10}$ cells.

Figure 9.5: *Relative compression error given as a function of the compression rate for two refinement thresholds:* $\xi_{refine} = 5.0 \cdot 10^{-2}$ *and* $\xi_{coarsen} = 2.5 \cdot 10^{-2}$ *in blue ;* $\xi_{refine} = 1.0 \cdot 10^{-2}$ *and* $\xi_{coarsen} = 5.0 \cdot 10^{-3}$ *in green. First-order* Relax-B *scheme.*



(a) $\xi_{\text{refine}} = 5.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 2.5 \cdot 10^{-2}$.



(b) $\xi_{\text{refine}} = 1.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 5.0 \cdot 10^{-3}$

Figure 9.6: *Advection test case on different AMR grids (defined by different compression levels L_{comp}). L_1 error comparing with the analytical solution. Second-order Relax-B scheme.*

Figure 9.7). The choice of the value of ξ_{refine} (an ξ_{coarsen}) has an effect on r_{comp}: a lower value for ξ_{refine} leads to less compressed grids.

Finally, Figure 9.8 presents the relation between r_{comp} and ϵ_{comp} for second-order solutions, with two different Δx_{min}, that are marked by dashed lines in Figure 9.6 and Figure 9.7. The difference with respect to the first-order case is that low values for ξ_{refine} do not ensure low ϵ_{comp} anymore: for very fine grids, the ϵ_{comp} seems even higher for lower values of ξ_{refine}. This observation seems to confirm that a refinement criterion based on the spatial first-order variations ($\Delta\rho$) of the solution is probably not adapted here.

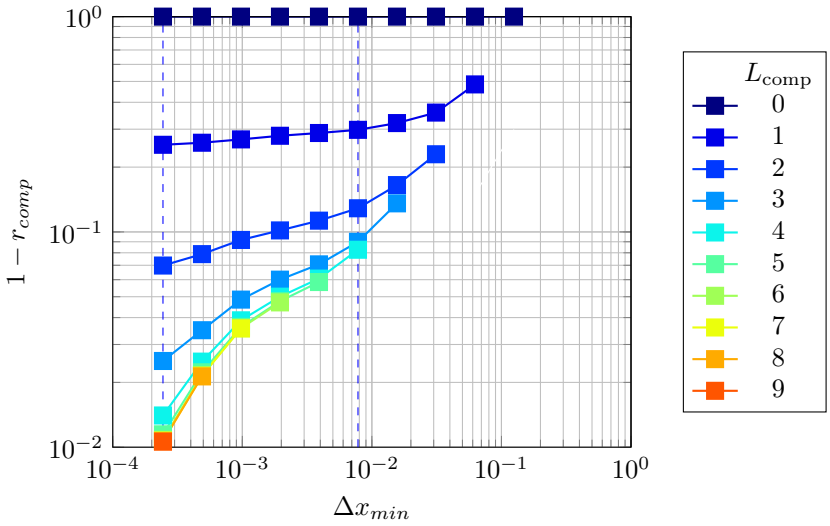
9.2.4 Conclusion

From the observations performed on the advection test solved with the first-order and second-order schemes, one can conclude that both the compression error and the compression rate are directly related with the values of the refinement thresholds. We have observed that increasing the value of these thresholds implies an increase of the compression error, which is detrimental for the accuracy of the numerical solution. However, increasing the value of the refinement thresholds increases the compression rate, which is interesting in terms of computation costs. This suggests that, in an AMR context, a balance between compression and accuracy needs to be reached. Nevertheless, one needs to take into account that in practice, AMR codes like CanoP, may not provide important performance gains when compared with fixed Cartesian grids codes, unless the compression rate is large enough. Let us also remark that effective gains of performance between the CanoP code and fixed grid codes is strongly dependent on the equations that are solved, as well as on the simulation case.

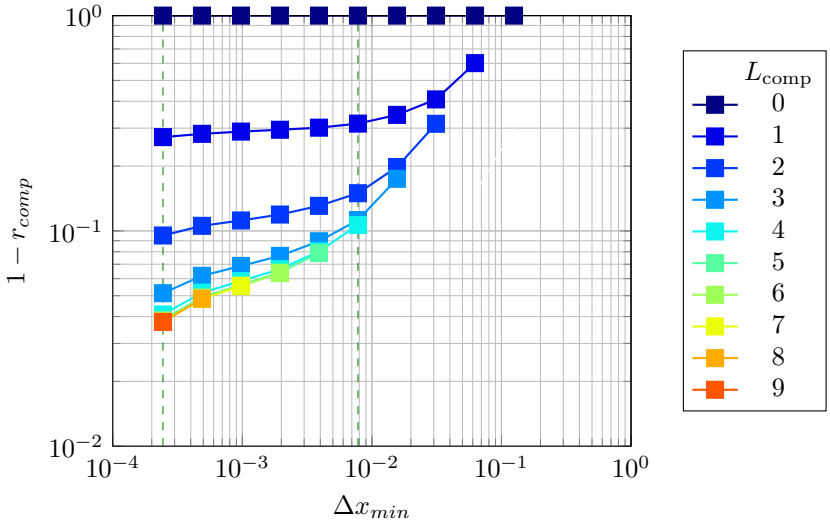
A good balance between compression and accuracy is also dependent on the numerical scheme, the test setting and the value of the refinement threshold, due to the use of heuristic refinement criteria. For this reason, we believe that the multi-resolution methods (Harten (1994); Müller (2003); Duarte (2012)) are probably good candidates for simulations that require to maintain the accuracy of the solution at a given rate.

Other elements may also have influenced the results presented in this section:

- when the grid is refined, there is no reconstruction of the solution in the new fine cells. While, this should not increase the truncation error when comparing with the former coarse grid, this could nevertheless represent a source of difference with a solution computed on a fine grid,
- we have noticed for some adapted solutions, that the truncation error was below the truncation error of the equivalent uniform grid. We believe that this could be the result of the time step choice, that is constrained by the maximum ratio between the wave velocities (acoustic and material velocities) and the cell size, evaluated in each cell of the grid. Acoustic wave velocities are often much larger than the matter velocity: in the case of the advection test computed on uniform grids, this yields a CFL condition on the time step which is very constraining and may cause numerical dissipation. However, for the advection test computed on adapted grids, the finest cells match regions where the acoustic velocity is low, i.e. mixture zones where the volume fraction α is not close to 0 nor 1, and Wood's sound velocity is much lower than the phasic sound velocities. Consequently, the constraint induced by the small space step is counterbalanced by the low acoustic velocities values. On the contrary, in regions where α is close to 0 or 1, the mesh gets coarser: the high sound velocity values are moderated by the large space steps in the CFL constraint. This feature of the AMR techniques could be further exploited with the help of local time-stepping methods.

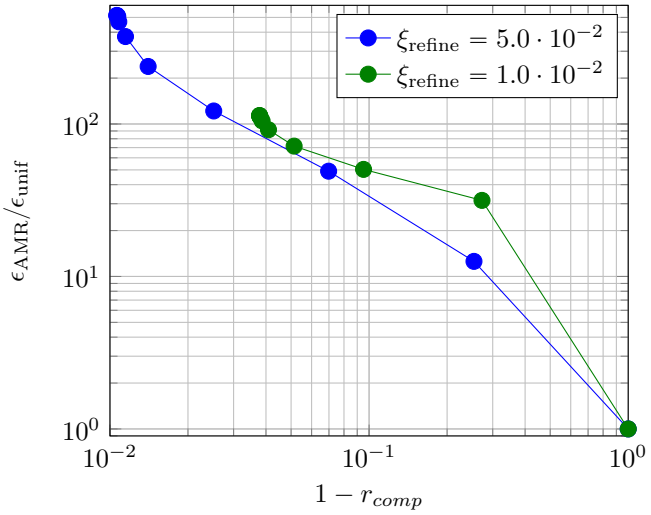


(a) Effective compression rates at the end of the simulation for $\xi_{\text{refine}} = 5.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 2.5 \cdot 10^{-2}$.

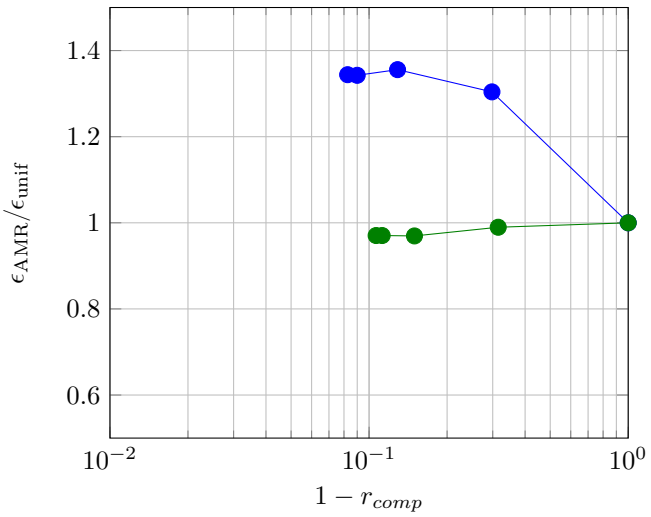


(b) Effective compression rates at the end of the simulation for $\xi_{\text{refine}} = 1.0 \cdot 10^{-2}$ and $\xi_{\text{coarsen}} = 5.0 \cdot 10^{-3}$.

Figure 9.7: *Advection test case on different AMR grids (defined by different compression levels L_{comp}). Compression rates. Second-order Relax-B scheme.*



(a) Equivalent uniform grid made of $2^{12} \times 2^{12}$ cells.



(b) Equivalent uniform grid made of $2^7 \times 2^7$ cells.

Figure 9.8: *Relative compression L_1-error given as a function of the compression rate for two refinement thresholds: $\xi_{refine} = 5.0 \cdot 10^{-2}$ and $\xi_{coarsen} = 2.5 \cdot 10^{-2}$ in blue ; $\xi_{refine} = 1.0 \cdot 10^{-2}$ and $\xi_{coarsen} = 5.0 \cdot 10^{-3}$ in green.*

9.3 1D Sod and rarefaction test cases

Now let us consider two classic Riemann problems for compressible flows: a shock tube (Sod shock tube) and a double rarefaction tests (see Toro (2009)). The numerical results are compared with the analytical solution (see Appendix A). Here, we intend to test the implementation of the numerical schemes in CanoPand the behavior of the AMR method with other refinement criteria and with the second-order schemes.

9.3.1 Presentation of the tests

Initial and boundary conditions:

These tests are performed in a 1D configuration, even on the 2D AMR grid. The initial conditions are given below.

Two-fluid shock initial condition:

	$\mathbf{W}_L$	$\mathbf{W}_R$
location	$x < 1$	$x > 1$
p	$1.1 \cdot 10^5$	$1.0 \cdot 10^5$
Y	$1.0 - 1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-11}$
$\mathbf{u}$	$\mathbf{0}$	$\mathbf{0}$
α	$1.0 - 1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-8}$

Double rarefaction initial condition:

	$\mathbf{W}_L$	$\mathbf{W}_R$
location	$x < 1$	$x > 1$
p	$1.0 \cdot 10^5$	$1.0 \cdot 10^5$
Y	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
$\mathbf{u}$	$(-1.0, 0.0, 0.0)^T$	$(1.0, 0.0, 0.0)^T$
α	0.5	0.5

The domain is made of two unit squares, so that it has the dimension 2×1 m. In the CanoP code, we then use two trees to get the whole domain. The boundary conditions are reflective. However, the choice of the final simulation time ensures that no wave involved in the Riemann problem has reached the domain borders.

Numerical scheme:

For these tests, we use Relax-B scheme applied to system (9.1). $\mathbf{g} = \mathbf{0}$ here. At second-order,we use the MUSCL-Hancock method. The reconstructed variables are the partial masses $m_1 = \rho Y$, $m_2 = \rho(1-Y)$, that must be positive (and then ensures the positivity of ρ, p and $0 < \alpha < 1$), and the velocity $\mathbf{u}$.

As far as the AMR methods are concerned, the refinement criterion is based involves the variations of α, p and $\mathbf{u}$. We recall here the formulae from Chapter 6 to compute the criterion $c(\mathbf{W})_i$ in cell K_i:

$$C(\mathbf{W})_i = \max_{j \leqslant 3} \left\{ D(\phi^{(j)})_i \right\},$$

with $\phi^{(1)} = \alpha$, $\phi^{(2)} = p$, $\phi^{(3)} = \mathbf{u}$, and:

$$D(\phi^{(j)})_i = \max \left\{ \left| \frac{\phi_k^{(j)} - \phi_i^{(j)}}{\max(\phi_i^{(j)}, \phi_k^{(j)})} \right|, k \in \mathcal{N}(i) \right\}.$$

This value is then compared with predefined refinement and coarsening thresholds ξ_{refine} and ξ_{coarsen}. For both tests, the minimum level of refinement is 1 ($\Delta x_{max} = 1/2$ m) and the maximum 8 ($\Delta x_{min} = 1/2^8$ m).

9.3.2 Numerical results for Sod test

An illustration of the AMR grid is given in Figure 9.9 for the second-order solution. One can see that the mesh is finer near the rarefaction, the contact discontinuity and the shock waves. However, due to some oscillations, it is also fine in a zone behind the shock wave. The grid is coarser out of these zones. One can note that the coarsest cells are not of size $\Delta x_{max} \times \Delta x_{max}$: due to the 2:1 balance constraint (see definition in Chapter 7) and the presence of fine cells in a large part of the domain, these large cells are not allowed to be used.

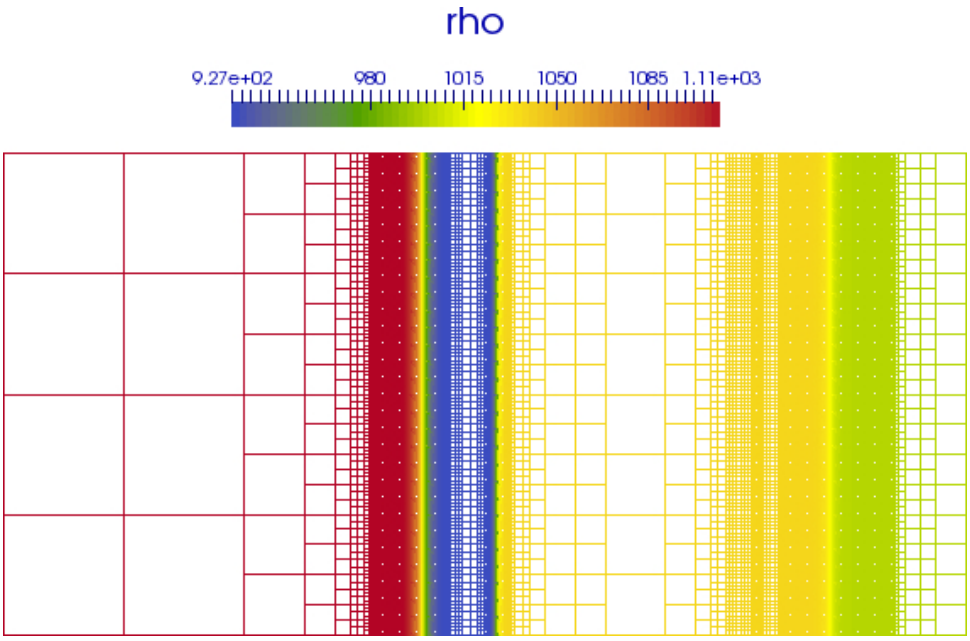


Figure 9.9: *Second-order Sod test case: illustration of the mesh and density profile at time* $t_f = 0.047s$.

Comparisons between the first-order and second-order solutions with the analytical solution are given in Figure 9.10 at time $t = 0.051$ s. We can see that, despite the small oscillations of the second-order solution, this solution is less diffusive than the first-order one, especially near the rarefaction and shock waves.

9.3.3 Numerical results for the double rarefaction test

The illustration of the AMR grid for the double rarefaction test is presented in Figure 9.11. Here, we can note that the whole zone between both rarefaction waves is at the finest level of refinement. This is explained by the variations of the velocity field in this zone, due to small amplitude oscillations. Indeed, although the magnitude of these velocity perturbations is very low (of the order of $1 \cdot 10^{-6}$), when considering the relative value from one cell to its neighbor, these relative variations can be at the order of 1. Consequently, the refinement criterion we use here is too sensitive to these variations and leads to the refinement of this zone. Here again, we can see the limits of using such a heuristic criterion.

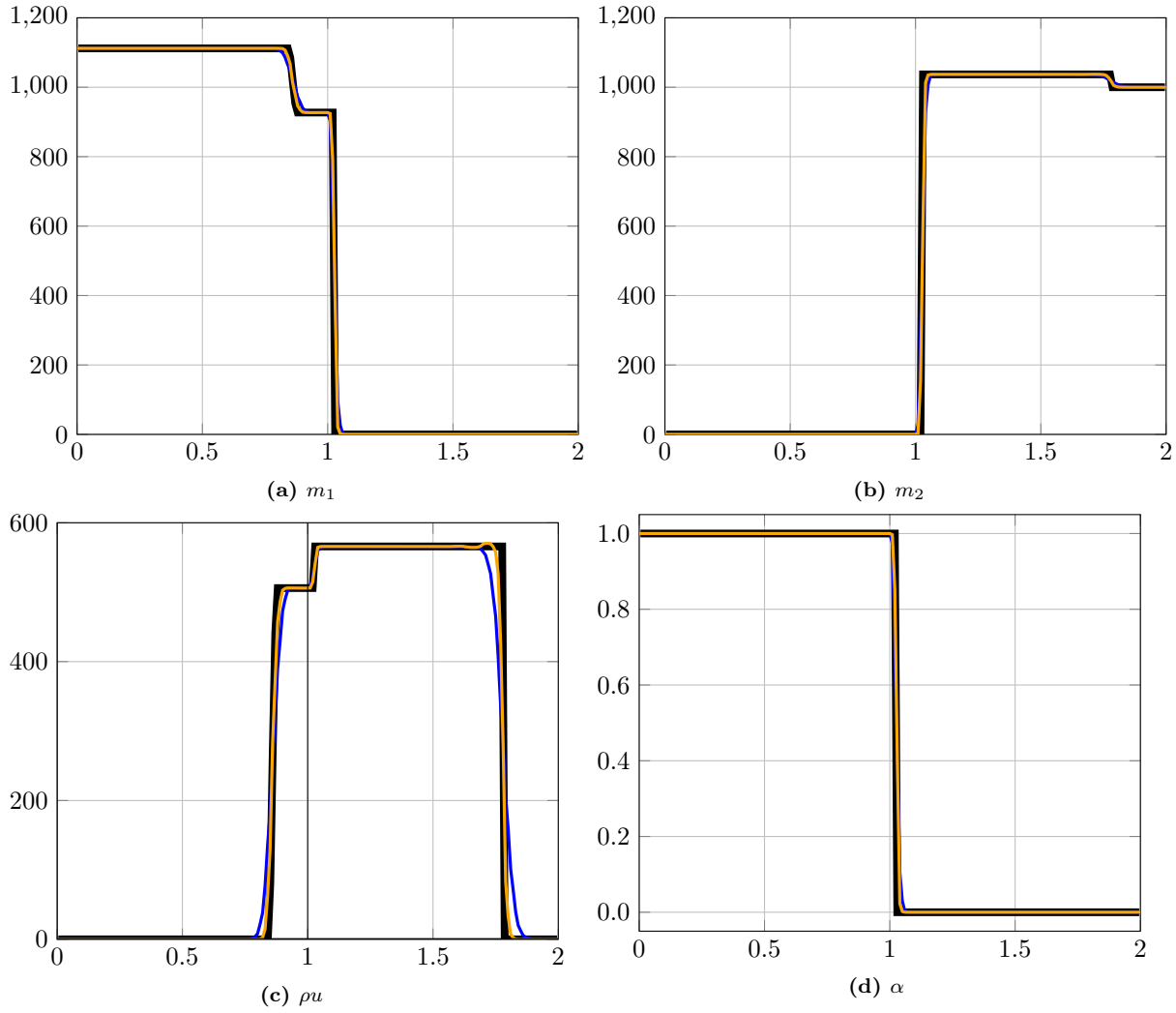
Comparisons between the numerical and the analytical solutions are given in Figure 9.12 at time $t = 0.024$ s. As for the Sod shock tube, we can see that the second-order solution is less diffusive than the first-order solution near the rarefaction waves.

9.3.4 Conclusion

With these first tests, we have verified that:

- the numerical methods, presented in Chapter 6, are compatible with AMR grids, and especially, one recovers the orders of convergence for the first-order and second-order schemes for the advection test case,
- these methods have been correctly integrated in the CanoP code, although some small-amplitude oscillations occur near shocks and rarefactions in the Sod and double rarefaction tests,
- the choices made for the refinement criteria permit to compress the mesh near the two-phase interface (for the advection case) and also near shocks and in the rarefaction zones.

As a practical conclusion, it appears that more sophisticated refinement criteria may improve the simu-



(a) m_1 **(b)** m_2 **(c)** ρu **(d)** α

Figure 9.10: *Sod test at time* $t = 0.051$ *s. First-order solution in blue. Second-order solution in orange.*

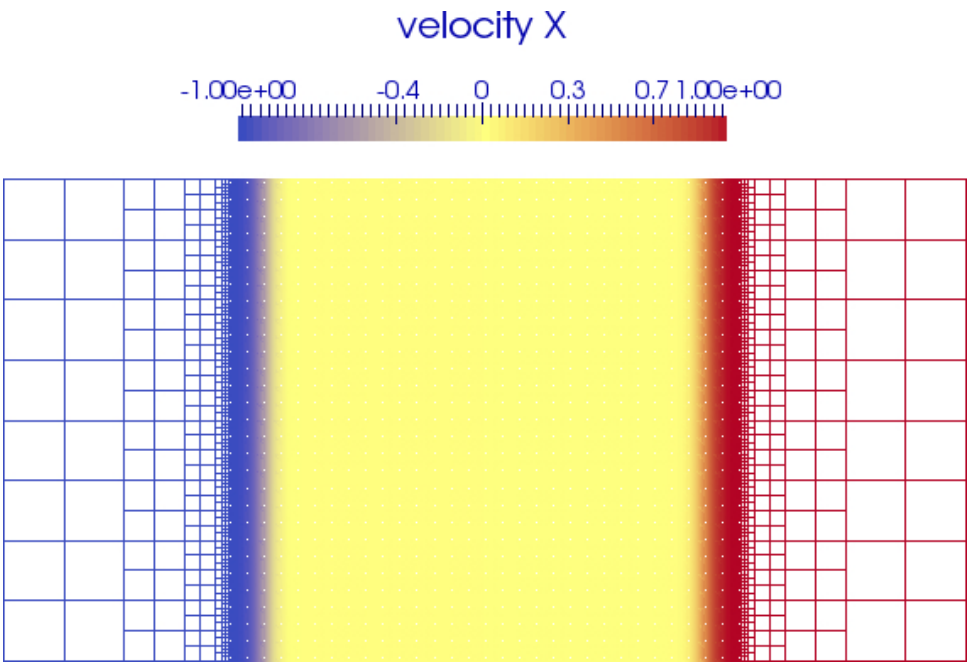
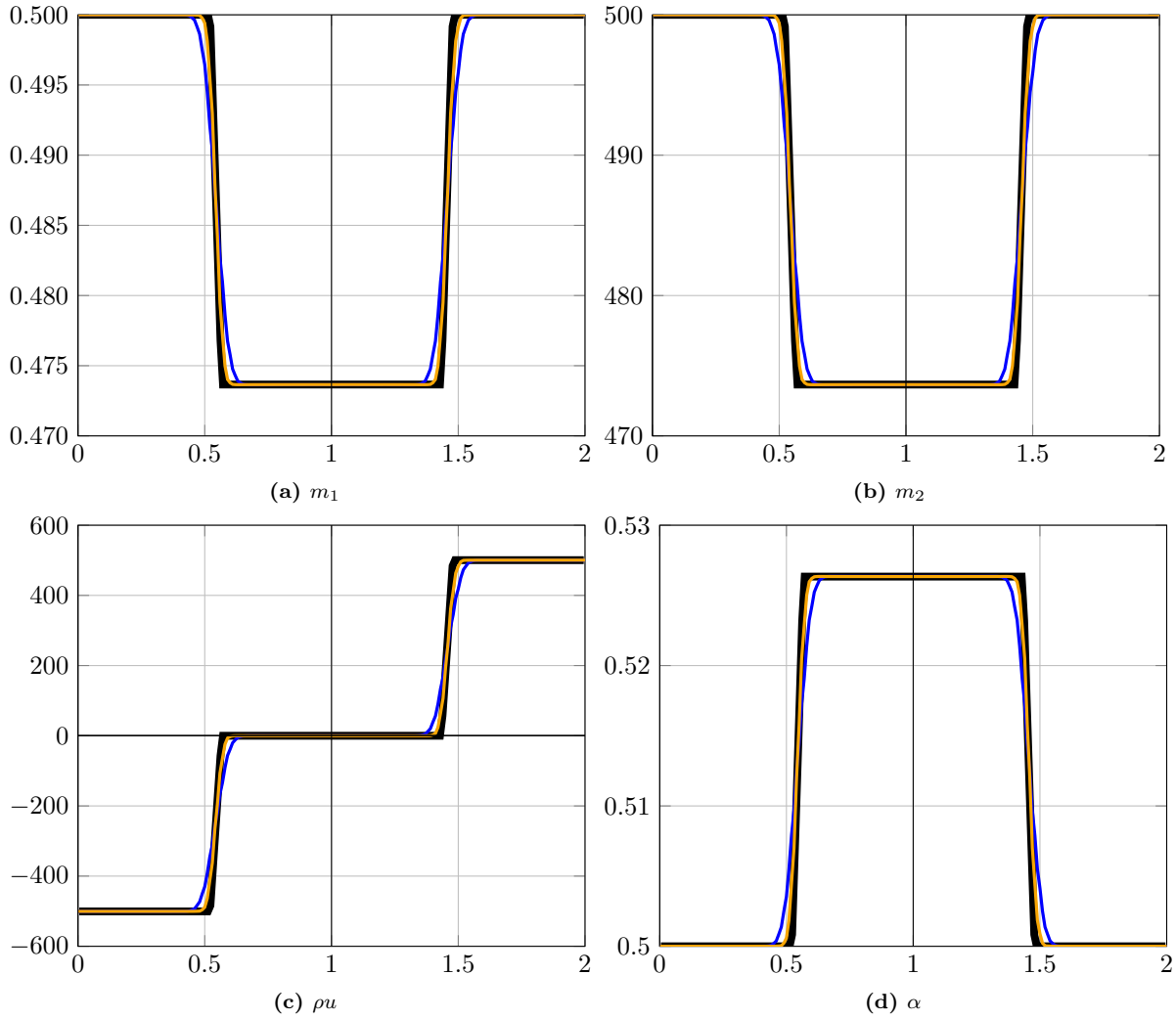


Figure 9.11: *Second-order double rarefaction test case: illustration of the mesh and velocity profile at time* $t_f = 0.024s$.

lations so that small-amplitude oscillations do not trigger unnecessary refinement, as we have observed in the double rarefaction case.



(a) m_1 **(b)** m_2 **(c)** ρu **(d)** α

Figure 9.12: *Double rarefaction test at time* $t = 0.024$ *s. First-order solution in blue. Second-order solution in orange.*

9.4 A 2D rising bubble test

Here we intend to test the AMR methods on a two-phase flow system with gravity. The flow will be governed by equations (9.1) with now $\mathbf{g} = (0, -g, 0)^T$, $g = 9.81\mathrm{m\,s}^{-2}$ (for practical reasons, in our tests, the gravity vector will be collinear with the second vector of the canonical basis).

9.4.1 Presentation of the test

The rising bubble test will give us the opportunity to study the behavior of our numerical strategies in a low-Mach number flow. Indeed, the initial condition of this test is a still gas bubble in a still liquid, so the whole velocity field is null. Then, due to gravity and the difference of density between the bubble and the environing liquid, the Archimedes' principle leads to a vertical motion of the bubble. In the first stages of this motion, the Mach number of in the liquid flow is below 0.01 (see in Figure 9.14 where the Mach number is computed at time $t = 1.4 \cdot 10^{-1}$ s with Relax-B scheme on a uniform triangular grid).

Initial and boundary conditions: The numerical initial condition for this test is given at any point $\mathbf{x} = (x, y)^T$ by:

$$\begin{cases} p(0,\mathbf{x}) &= f(y), \\ \alpha(0,\mathbf{x}) &= g(\mathbf{x}), \\ \mathbf{u}(0,\mathbf{x}) &= \mathbf{0}, \end{cases} \tag{9.3}$$

with:

$$f(y) = p^o + \begin{cases} c_l^2 \rho_l^o \left(\exp\left(-\frac{g(y-h_0)}{c_l^2} \right) - 1 \right) & \text{if} \quad y < h_0, \\ c_g^2 \rho_g^o \left(\exp\left(-\frac{g(y-h_0)}{c_g^2} \right) - 1 \right) & \text{else,} \end{cases}$$

where $h_0 = 0.9$, $c_g = 3.4 \cdot 10^2$, $\rho_g^o = 1.0$, $c_l = 1.5 \cdot 10^3$, $\rho_l^o = 1.0 \cdot 10^3$ $p^o = 1.0 \cdot 10^5$, $g = 9.81$, and

$$g(\mathbf{x}) = \begin{cases} 1.0 & \text{if} \quad y > h_0, \\ 1.0 & \text{else if} \quad |\mathbf{x} - \mathbf{x}_0| < r_0, \\ 0.0 & \text{else,} \end{cases} \quad \text{with} \quad \begin{cases} \mathbf{x}_0 &= (0.5, 0.2)^T, \\ r_0 &= 0.1. \end{cases}$$

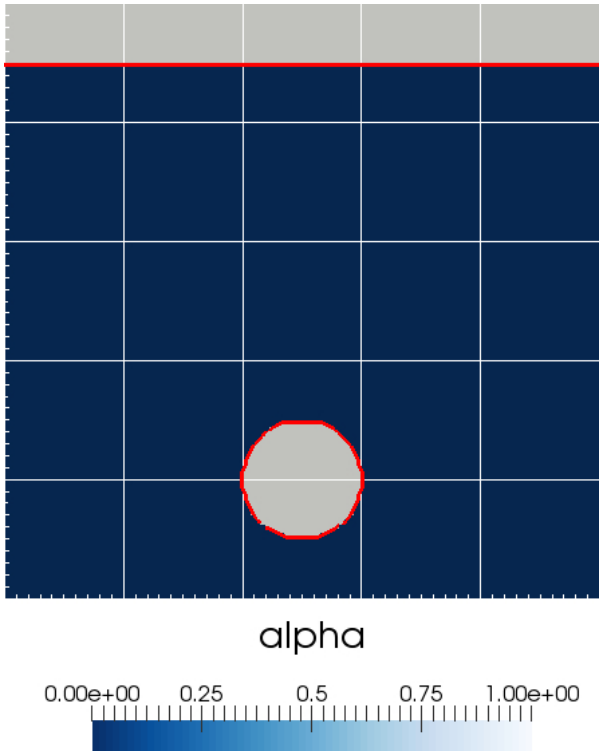
The boundary conditions are reflective on all sides of the domain.

Illustrations of the volume fraction and pressure profiles are given on Figure 9.13. Concerning the initial conditions, let us remark that:

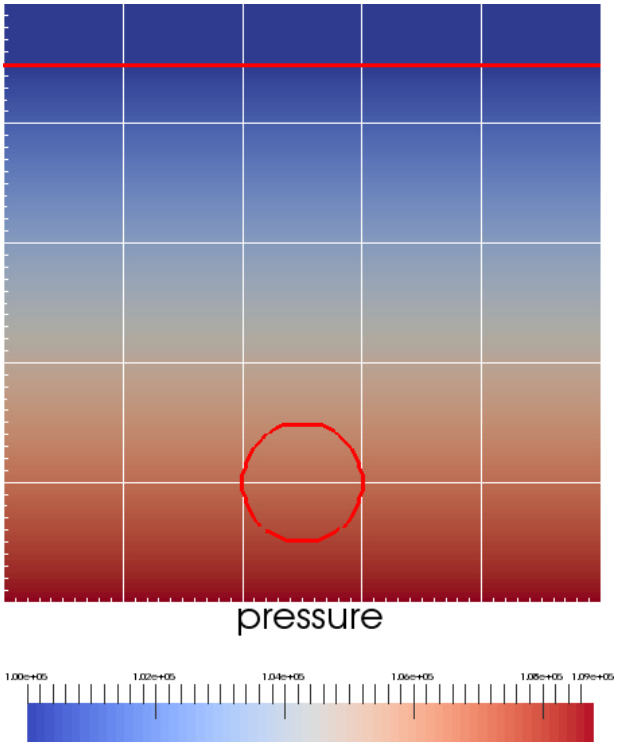
- we assume that there is a free surface in the upper part of the domain. The reason for this choice will be illustrated in the following, by comparing the numerical schemes. Moreover, this free surface changes the effective boundary that is seen by the liquid phase: the boundary conditions in the liquid phase are then reflective on the vertical borders and on the bottom border, while there is an interaction with the gas phase at the upper border,
- initially, there cannot be a hydrostatic equilibrium around the bubble, due to Archimedes' force. It is then complex to define the exact initial condition. The choice that has been made here is to consider that the same pressure profile applies in and outside the bubble. This initially hydrostatic profile is given by the pressure profile in a pure liquid.

The numerical solutions of this test may not be physically relevant. Indeed, multiple effects are not modeled, such as the liquid viscosity and the surface tension. They may have an influence on the bubble's shape and position through time, although the bubble initial dimension is quite large: its radius is 20 cm.

Another probable cause of error comparing with a physical solution lies in the definition of the initial condition. Indeed, the initial hydrostatic profile that applies to the liquid phase should not be present inside the bubble, on a one hand due to surface tension effects, and on the other hand as the density



(a) Volume fraction profile on a 128 × 128 uniform grid.



(b) Pressure profile on a 128 × 128 uniform grid.

Figure 9.13: *Initial condition for the rising bubble test case. The red curve stands for the* $\alpha = 0.5$ *iso-line.*

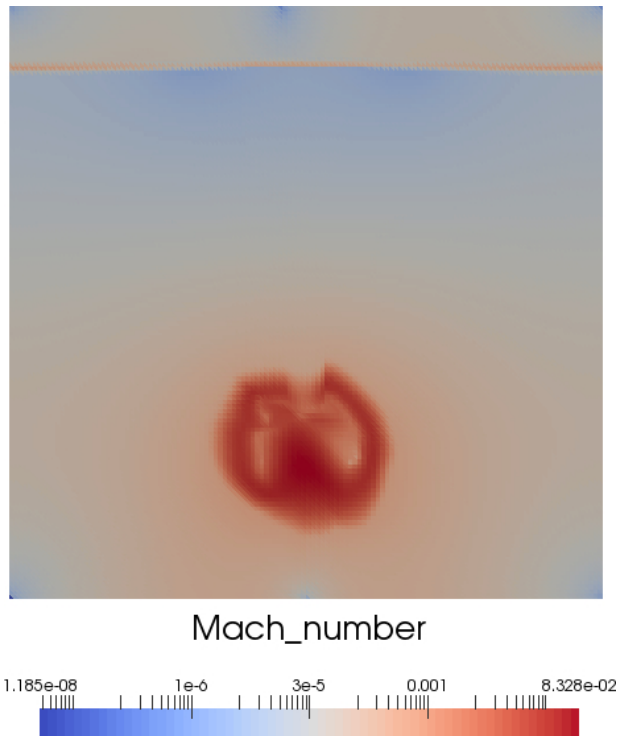


Figure 9.14: *Mach number value at time* $t = 1.4 \cdot 10^{-1}$ *s, solution from Relax-B scheme on a uniform triangular grid made of* $2 \times 128 \times 128$ *cells.*

gradient is different between gas and liquid. However, we expect this defect to be rapidly damped during the numerical simulation, due to pressure balancing.

Finally, the numerical viscosity at the bubble interface leads to zones of gas and liquid mixing, that is not observed in a physical experiment, and may change the bubble acceleration.

To complete this short discussion, let us say that the results that are presented hereafter, although not compared with experiments, will enable to show the difficulties of the rising bubble test case, that are: the low-Mach flow configuration and AMR grids.

Numerical schemes: In the following, we will use the Relax-B scheme applied to system (9.1). The same scheme with the all-regime modification of Chalons, Girardin, and Kokh (2016) and presented in Chapter 6, section 6.5, will be noted here Relax-BM. In all following simulations, there is no dimensional splitting: it is indeed not convenient with the triangular grid, and we have noted that it is not compatible with the all-regime scheme. However, we use an operator splitting to integrate the gravity source term. The CFL condition is taken to be 0.5. The numerical solutions are obtained with first order schemes: the second-order reconstruction could not be tested as it has not been implemented with Relax-BM yet, nor on triangular grids. Finally, when using adapted grids, the refinement criterion is $\Delta\alpha$, $\xi_{\text{refine}} = 1.0 \cdot 10^{-5}$ and $\xi_{\text{coarsen}} = 5.0 \cdot 10^{-6}$.

9.4.2 Comparison of shape and position of the bubble

In this paragraph, we will use the following nomenclature to indicate the different numerical solutions:

Name	Scheme	Grid	L_{min}	L_{max}
BQ_x-X	Relax-B scheme	quadrangular	x	X
LMQ_x-X	Relax-BM scheme	quadrangular	x	X
BT_x-X	Relax-B scheme	triangular	x	X

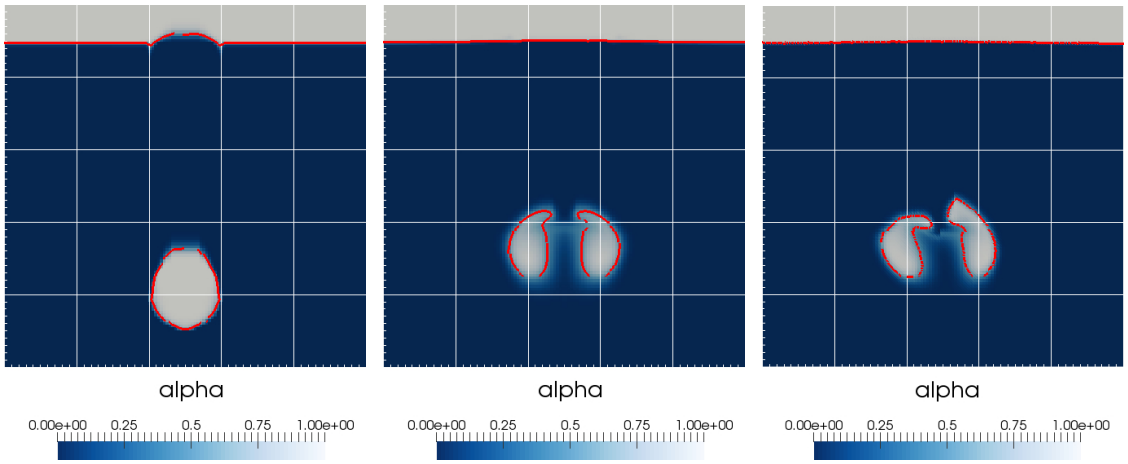
Uniform grids

Let us first observe the effects of the low-Mach flow on uniform grids. As we can see on Figure 9.15, the differences between the numerical solutions are noticeable:

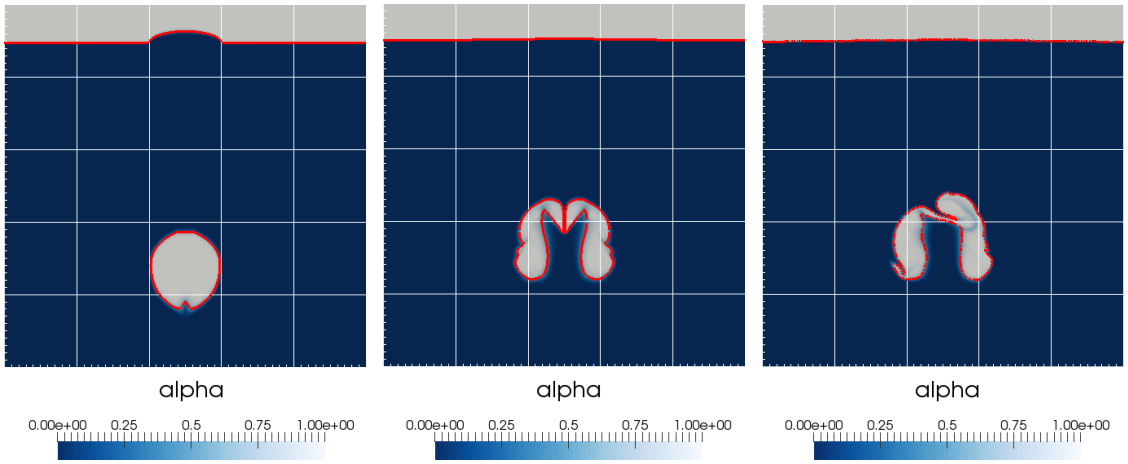
- the position of the bubble at the final time differs between the solutions BQ_7-7 and both solutions LMQ_7-7 and BT_7-7. Indeed, the bubble in BQ_7-7 solution has moved over a shorter distance than the bubbles LMQ_7-7 and BT_7-7.
- the form of the bubble also differs. While it remains close to a sphere in BQ_7-7, it seems to have split in LMQ_7-7 and BT_7-7. Moreover, in BT_7-7, the shape is deformed and not symmetric (due to the orientation of the triangles),
- we can note also that the free surface also deforms. This deformation is located above the bubble only, while in LMQ_7-7 and BT_7-7, the deformation has spread over the whole surface,
- for solutions that were obtained on finer grids, we can note that the position of the bubble in BQ_10-10 is higher than in BQ_7-7. By using a finer grid, we probably have removed some numerical viscosity, that was preventing the bubble from rising in BQ_7-7. However, it is not yet at the same height as in LMQ_10-10 and BT_10-10,
- the shape of the bubble in LMQ_10-10 and BT_10-10 is much more deformed with this level of fine grid,
- with the finest grids, numerical diffusion has also been reduced at the gas-liquid interface in all simulations.

Adapted grids

The same comparisons as in the previous paragraph are now performed on adapted grids. For these grids, the minimum level of refinement is 7 ($\Delta x_{max} = 1/2^7$ m), which corresponds to the coarse uniform

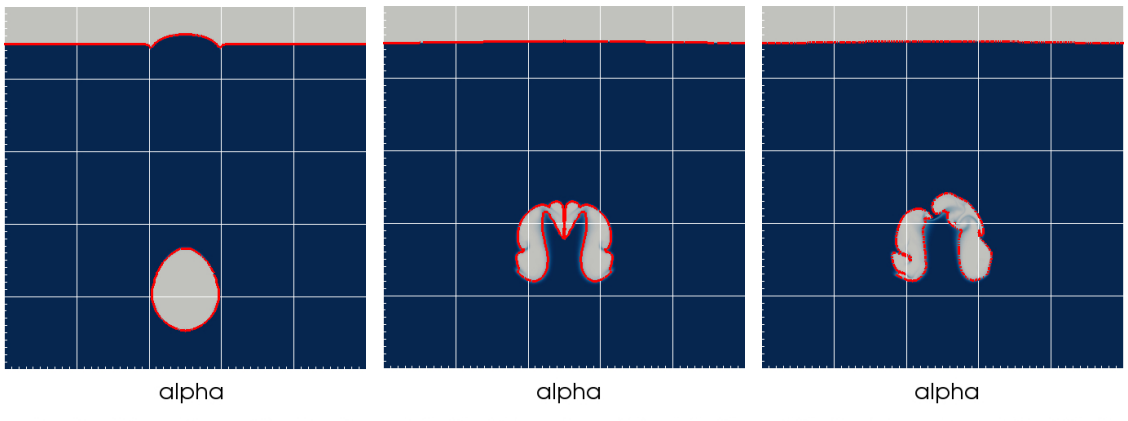


(a) BQ_7-7 solution. **(b)** LMQ_7-7 solution. **(c)** BT_7-7 solution.



(d) BQ_10-10 solution. **(e)** LMQ_10-10 solution. **(f)** BT_10-10 solution.

Figure 9.15: *Rising bubble test: volume fraction α profile of the different numerical solutions at time $t_f = 0.28s$. The red curve stands for the $\alpha = 0.5$ iso-line.*



(a) BQ_7-11 solution. **(b)** LMQ_7-11 solution. **(c)** BT_7-11 solution.

Figure 9.16: *Rising bubble test: volume fraction α profile of the different numerical solutions at time $t_f = 0.28s$. The red curve stands for the $\alpha = 0.5$ iso-line.*

grid of the previous paragraph, and the maximum level is 11 ($\Delta x_{min} = 1/2^{11}$ m), which is one level finer than the finest grid of the previous paragraph. Comparisons of the solutions BQ_7-11, LMQ_7-11 and BT_7-11 are presented in Figure 9.16. One can note again differences between Relax-B and Relax-BM:

- the bubble is located at a lower position with BQ_7-11 than with LMQ_7-11 or BT_7-11,
- the bubble shape obtained with LMQ_7-11 and BT_7-11 is strongly deformed. This deformation is once more not symmetrical with BT_7-11.

Now, by comparing with the solution of uniform grids, one can note few differences between LMQ_7-11 and LMQ_10-10 or BT_7-11 and BT_10-10, apart from lower numerical diffusion with the adapted grid, because of its highest level of refinement. However, when considering the location of the bubble, solution BQ_7-11 is closer to solution BQ_7-7 than to solution BQ_10-10. The refinement criterion, based on $\Delta\alpha$ does not seem to help improving the solution obtained on quadrangular grids with the Relax-B scheme. We will see in the following paragraph, with a study of the velocity field, that the grid has an influence on $\mathbf{u}$.

9.4.3 Comparison of the velocity fields

General velocity field: As we have noted in the previous paragraphs, the dynamics of the bubble is significantly dependent on the discretization strategy. The study of the resulting velocity fields sheds even more light on this matter.

In Figure 9.17, one can note that the velocity fields of solutions BQ_7-7 and BQ_10-10 seem to be oriented along the vertical and horizontal axes, while for LMQ_7-7 and BT_7-7, it is more isotropic. The liquid that is pushed by the moving bubble is moving vertically in BQ_7-7, up to the liquid surface. This explains the apparition of this surface deformation above the bubble. On the contrary, in LMQ_7-7 and BT_7-7, the volume of liquid that is displaced by the bubble, is spread in all directions, and then goes back below the bubble. One can see that this motion is partially present with BQ_10-10: refining the grid leads to the same form of velocity field as for LMQ_7-7 and BT_7-7.

Remark 38. *This kind of velocity fields reminds us of the Hodge-Helmholtz decomposition analysis that was mentioned in Chapter 6: in the low-Mach limit, solutions of Godunov-type schemes on Cartesian grids lie in a space of solutions where $\partial_x u_x = 0$ and $\partial_y u_y = 0$ in 2D. However, if the numerical viscosity is decreased, as in LMQ_7-7 or BQ_10-10, then the solution tends to the divergence-free velocity field $div_{\mathbf{x}}\mathbf{u} = 0$, that is characteristic of incompressible flows.*

Finally, the non-symmetric deformation of the bubble in BT_7-7 is also visible in its velocity field: we can see that pieces of the liquid phase move with strong velocity magnitudes in the opposite direction of the mean fluid motion.

Checkerboard modes due to the all-regime scheme: Although the solution LMQ_7-7 obtained with the all-regime scheme seems to give more relevant numerical results, some difficulties are encountered with this method. One of them is related to the apparition of *checkerboard* modes in the velocity and pressure fields (see also Dellacherie (2009)). This phenomenon is illustrated in Figure 9.18a in the gas phase. According to the configuration, these spurious variations may have a non-negligible effect on the numerical solution. They are probably due to the lack of numerical viscosity, and cures to this defect are still an open research field.

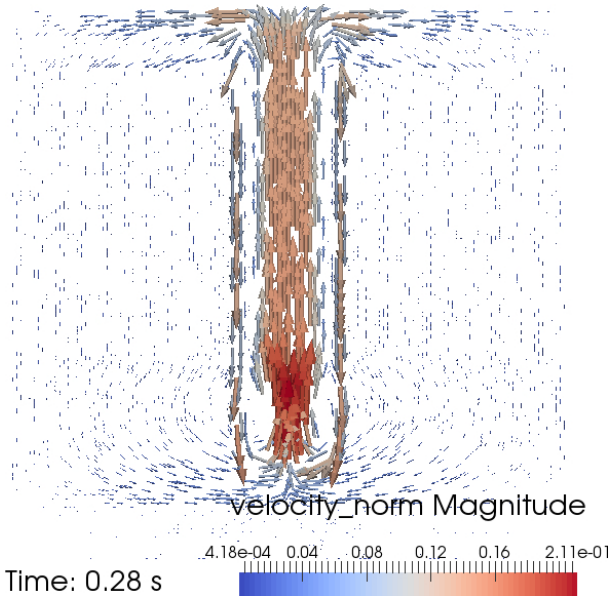
Pressure waves due to AMR: The other difficulty that can be encountered in compressible flows with gravity seems to be typical of AMR grids, and is illustrated in Figure 9.18b with a triangular mesh (but is also present in quadrangular meshes with the low-Mach correction). We can indeed see a sort of pressure wave, generated just below the bubble, at the frontier of fine and coarser cells. This comes from an overshoot in the pressure profile just after the refinement or the coarsening of local cells. No solution has yet been found to tackle this issue. Different procedures for reconstructing the variables data when refining or coarsening have been tested, but these pressure defects are still generated. However, for the simulations that were performed up to now, this problem did not show to have a significant impact on the final numerical solution.

9.4.4 Conclusion

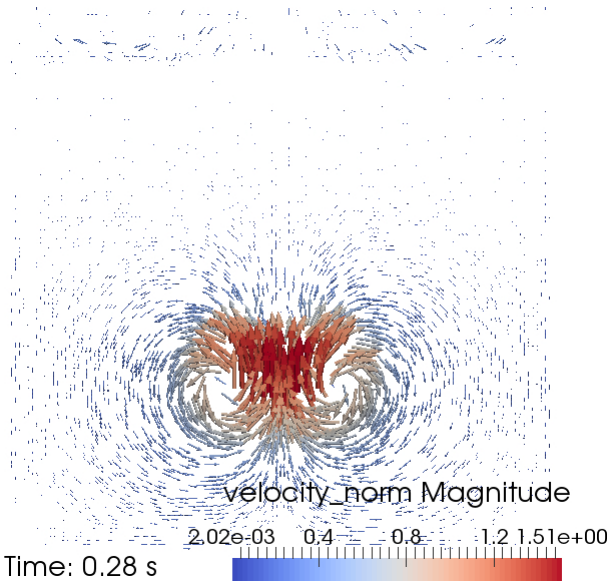
The test case of the rising bubble clearly shows the difficulties related to the low-Mach number regime when using compressible equations in association with classic Godunov-type schemes. Without any scheme modification, the solution that is computed on adapted grids, although being less diffusive at the two-phase interfaces, seems to have the same dynamics as a solution computed on a coarse grid. The flow dynamics is then improved when a special treatment for the low-Mach regime is used, i.e. either the all-regime scheme of Chapter 6 or the adapted triangular grid. However, some other problems appear:

- the bubble suffers strong deformations that depend on the grid choice. First tests, consisting in artificially spreading the bubble interface at the initial condition, were performed and seemed to show an improvement in the shape of the bubble while preserving the same dynamics. However, these tests have not been thoroughly studied and are not presented here,
- the scheme with the all-regime modification generates pressure and velocity profiles that are known as checkerboard modes. This defect does not alter the solution too much for the bubble test case, but a deeper study would probably show its real influence,
- while the checkerboard modes are not specifically associated with adapted grids (they have been observed on Cartesian grids too), the problem of the pressure waves, that occurs with Relax-BM scheme and on the triangular grid, seems to be specifically due to the refining and coarsening processes.

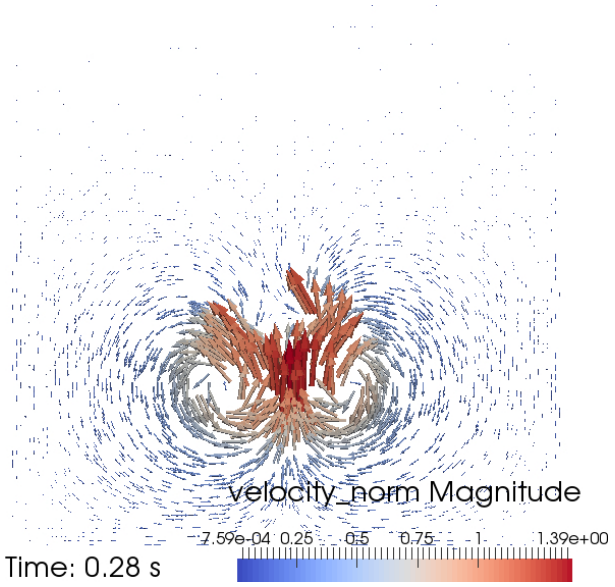
These observations open new axes of research to better understand the structure of the numerical solutions.



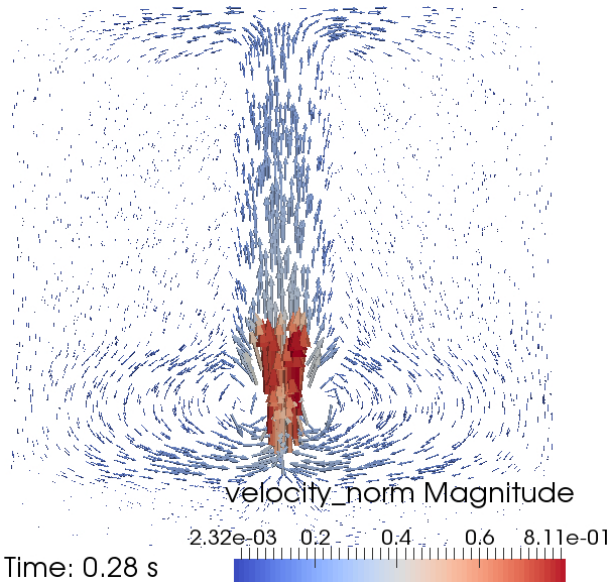
(a) Relax-B scheme on a uniform 128 × 128 grid.



(b) Relax-BM scheme on a uniform 128 × 128 grid.

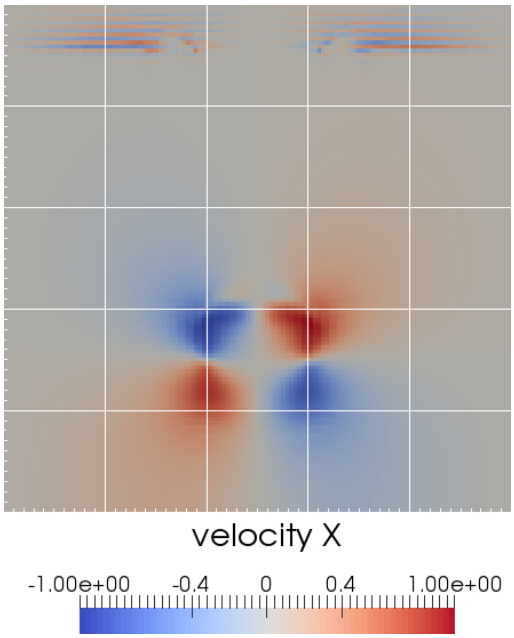


(c) Triangular 2 × 128 × 128 grid.

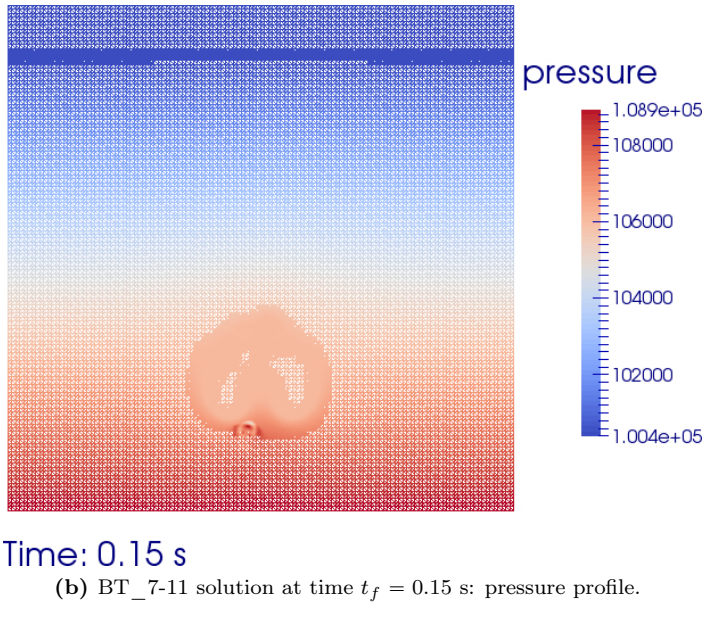


(d) Relax-B scheme on a uniform 1024 × 1024 grid.

Figure 9.17: *Rising bubble test: velocity field at time* $t_f = 0.28$. *The colormaps are different from one solution to the other.*



(a) LMQ_7-7 solution at time $t_f = 0.28$ s: horizontal velocity component.



(b) BT_7-11 solution at time $t_f = 0.15$ s: pressure profile.

Figure 9.18: *Rising bubble test: issues of the low-Mach correction (upper figure) and AMR solutions.*

9.5 Dam break with an obstacle

The present dam break simulation will help us to test the ability to use a forest of trees with p4est. The setting of this test matches the experience of Koshizuka, Tamako, and Oka (1995), and has been considered as a numerical test in several works (see in Golay, Ersoy, Yushchenko, and Sous (2015); Bernard-Champmartin and De Vuyst (2014), for instance). Although experimental data is available, here only qualitative comparisons with the experimental results are performed. In Chapter 10, we will present a more thorough study of the dam break problem.

Initial and boundary conditions: The numerical initial condition for this test is given at any point $\mathbf{x} = (x, y)^T$ by:

$$\begin{cases} p(0,\mathbf{x}) &= p^o, \\ \alpha(0,\mathbf{x}) &= g(\mathbf{x}), \\ \mathbf{u}(0,\mathbf{x}) &= \mathbf{0}, \end{cases} \tag{9.4}$$

with $p^o = 1.0 \cdot 10^5$ and

$$g(\mathbf{x}) = \begin{cases} 0.0 & \text{if} \quad y < h_0 \text{ and } x < l_0, \\ 1.0 & \text{else,} \end{cases} \qquad \text{with} \qquad \begin{cases} h_0 &= 0.292, \\ l_0 &= 0.146. \end{cases}$$

The boundary conditions are reflective on all the borders of the domain. The dimensions of the domain, as well as the initial condition are represented in Figure 9.19.

Due to the dimensions of the domain and to get a conforming initial macro-mesh, a forest of 498 trees is created (using Gmsh as in Appendix E). It is illustrated, as well as the grid at initial time in Figure 9.20. The other figure of Figure 9.20 shows the partition of the domain over 120 MPI processes on the mesocenter of CentraleSupelec. The total computation time was about 10 hours.

Numerical scheme: The numerical solution is obtained with Relax-B first-order scheme applied to (9.1). In order to discard possible low-Mach issues, we use a triangular mesh, as displayed on Figure 9.20.

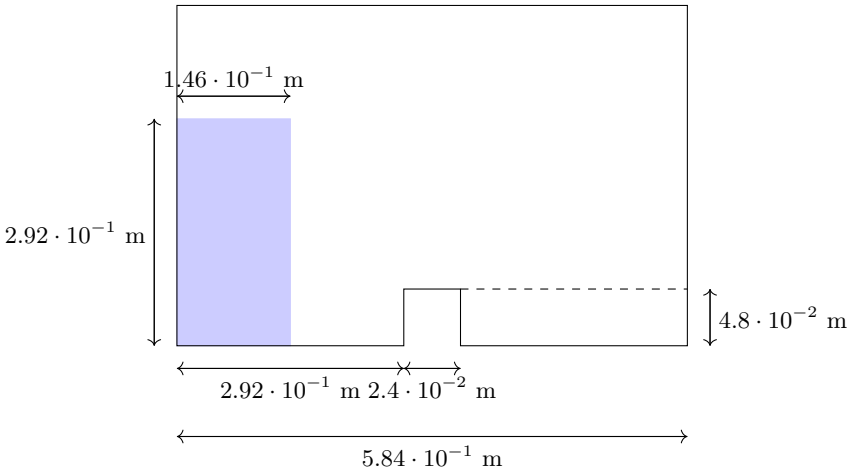


Figure 9.19: *Dimensions and initial condition for the test of the dam break with obstacle.*

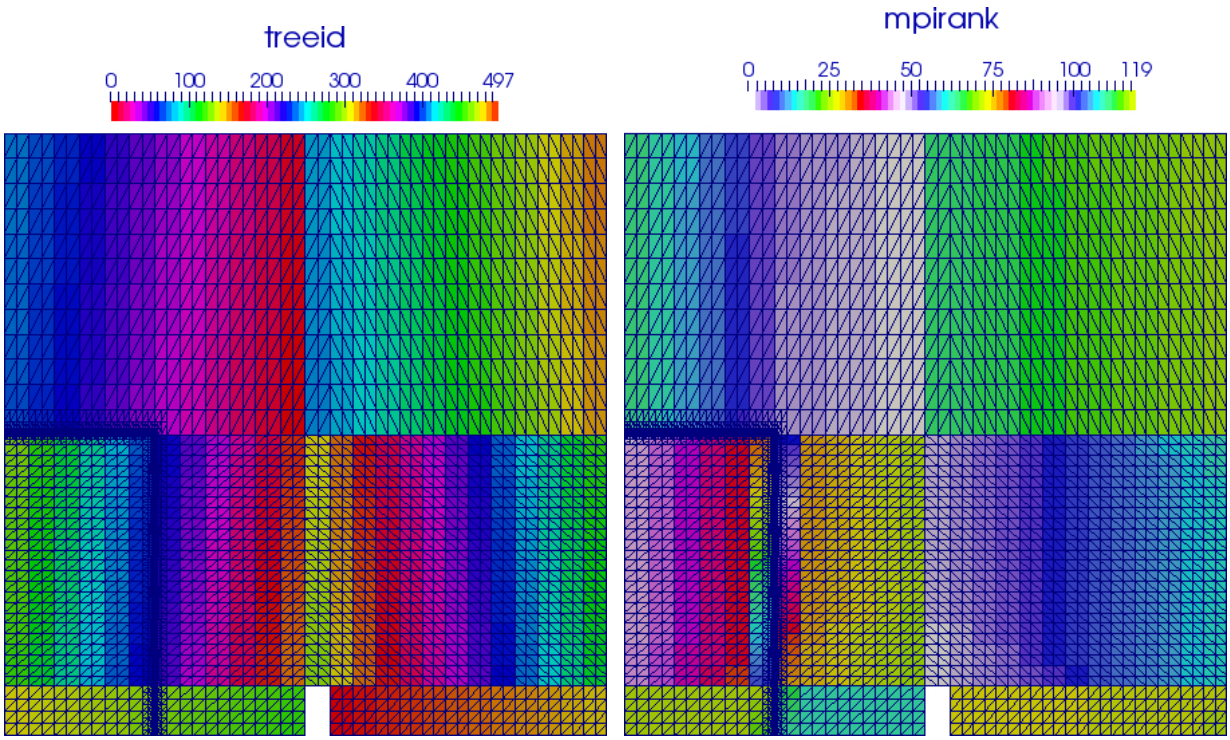


Figure 9.20: *Forest of trees and MPI partition at the initial time of the dam break with obstacle test.*

Time	N_{tri}	r_{comp}
0.00 s	$1.1 \cdot 10^4$	0.99
0.20 s	$1.8 \cdot 10^5$	0.82
0.30 s	$3.3 \cdot 10^5$	0.67
0.40 s	$6.4 \cdot 10^5$	0.38
0.50 s	$7.9 \cdot 10^5$	0.22

Table 9.1: *Number of (triangular) cells N_{tri} and compression rate r_{comp} of the grid for the dam break with obstacle test, at different instants.*

The minimum level of refinement in each of the 498 trees is 1, and the maximum level is 5. The refinement criterion is based on the gradient of the volume fraction α, with refinement thresholds $\xi_{\text{refine}} = 5.0 \cdot 10^{-6}$ and $\xi_{\text{coarsen}} = 1.0 \cdot 10^{-6}$.

Visual comparisons with the experimental results: In Figure 9.21, we present a qualitative comparison between the numerical solution and experimental photographs, issued from Golay, Ersoy, Yushchenko, and Sous (2015). We make comparisons at four different instants:

- $t = 0.2$ s: liquid has flowed through the domain and reached the obstacle, generating a liquid wave over the obstacle. The simulated wave seems to be a bit finer than the experimental one, but both its shape and location are coherent with the experimental observations,
- $t = 0.3$ s: the wave is moving forward in the direction of the opposite wall. Here again, we can note a difference in the shape of the wave between the numerical and the experimental solutions,
- $t = 0.4$ s: the wave has reached the opposite wall for both numerical and experimental solutions. The impact of the liquid wave onto the wall seems well represented by the simulation results.
- $t = 0.5$ s: the big wave breaks up and starts to fall back behind the obstacle. In particular, one can note that a second wave is forming close to the obstacle in the experiment. This wave is also present in the numerical solution.

The solution seems quite spread by the numerical diffusion: the liquid-gas interface is significantly diffused, leading to the refinement of many cells. Thus, as we can see Table 9.1, the compression rates for this simulation are in fact low and decrease through time.

Conclusion: With this first dam break case, we have seen that, by using p4est, we can handle a forest of numerous trees and perform simulations on domains that are more complex than simple unit cubes. Let us mention that the triangular grid feature of CanoP has only been tested with first-order schemes. This results in important numerical diffusion in our simulations. The compression rate of the adapted grid is consequently decreasing during the computation and we gradually loose the advantages of the AMR grid. An effort should be done here to limit the numerical diffusion at the two-phase interface, by using higher-order schemes or low-diffusive ones as in Kokh and Lagoutière (2010); Bernard-Champmartin and De Vuyst (2014). The simulation results are nevertheless satisfying and seem quite coherent with the photographs of the experiment.

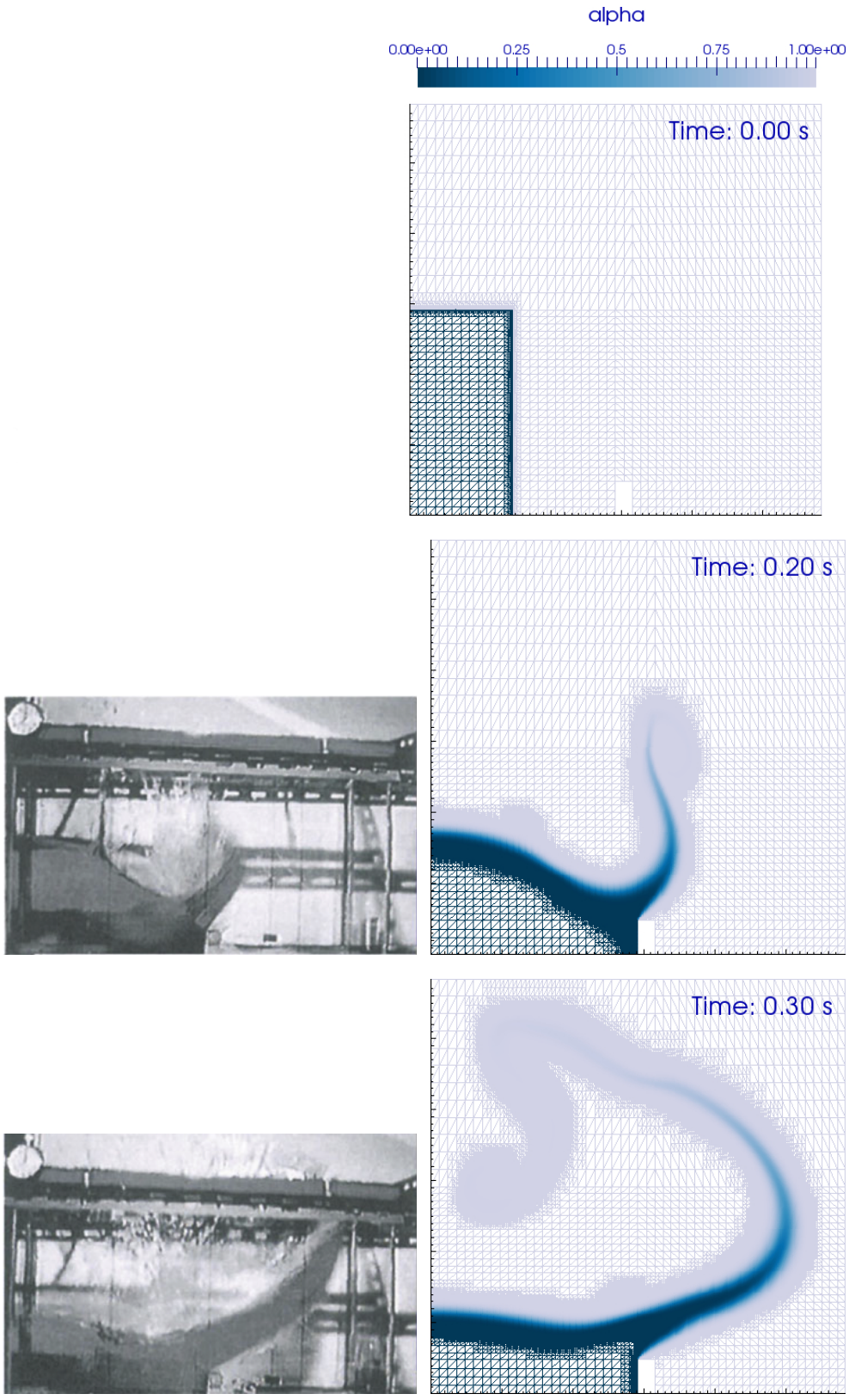


Figure 9.21: *Experimental results (left column, pictures from Golay, Ersoy, Yushchenko, and Sous (2015)) and numerical solution (right column) for the dam break problem with obstacle at different times.*

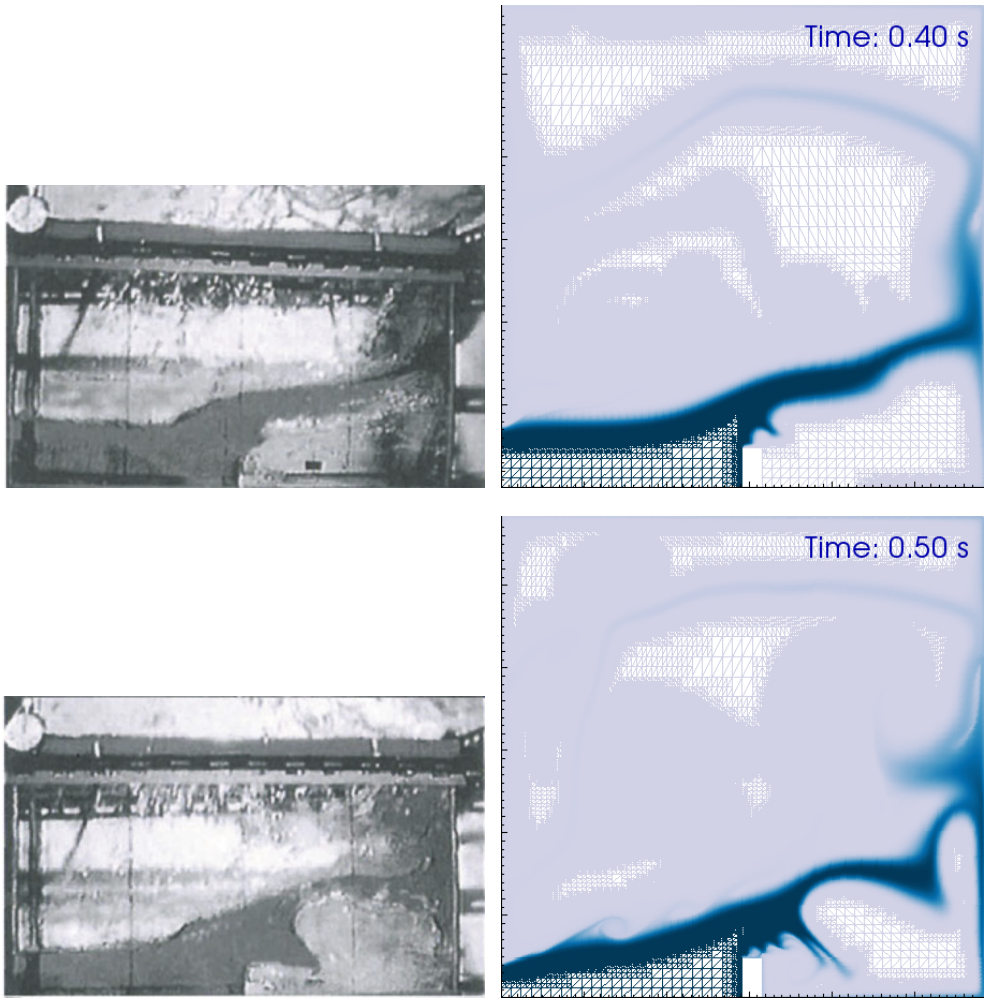


Figure 9.21: *(continued) Experimental results (left column, pictures from Golay, Ersoy, Yushchenko, and Sous (2015)) and numerical solution (right column) for the dam break problem with obstacle at different times.*

9.6 3D dam break simulation

Few 3D configurations are presented in this chapter. Let us recall that a 3D advection test was used to get the performance results of CanoP in Chapter 8. Here, we present a simple 3D dam-break simulation, performed in a unit cube.

Initial and boundary conditions: The numerical initial condition for this test is given at any point $\mathbf{x} = (x, y, z)^T$ by:

$$\begin{cases} p(0,\mathbf{x}) &= f(\mathbf{x}), \\ \alpha(0,\mathbf{x}) &= g(\mathbf{x}), \\ \mathbf{u}(0,\mathbf{x}) &= \mathbf{0}, \end{cases} \tag{9.5}$$

with:

$$f(\mathbf{x}) = p^o + \begin{cases} c_l^2 \rho_l^o \left(\exp\left(-\frac{g(y-h_0)}{c_l^2} \right) - 1 \right) & \text{if} \quad g(\mathbf{x}) = 0.0, \\ c_g^2 \rho_g^o \left(\exp\left(-\frac{g(y-h_0)}{c_g^2} \right) - 1 \right) & \text{else,} \end{cases}$$

where $c_g = 3.4$, $\rho_g^o = 1.0$, $c_l = 15.0$, $\rho_l^o = 1.0 \cdot 10^3$ $p^o = 1.0 \cdot 10^5$, $g = 9.81$, and

$$g(\mathbf{x}) = \begin{cases} 0.0 & \text{if} \quad y < h_0 \text{ and } x, z < l_0, \\ 1.0 & \text{else,} \end{cases} \qquad \text{with} \qquad \begin{cases} h_0 &= 0.8, \\ l_0 &= 0.4. \end{cases}$$

The boundary conditions are reflective on all sides of the domain.

Numerical scheme: Here, exceptionally we use Relax-S scheme without any modification, with equations (9.1) with $\mathbf{g} = (0, -9.81, 0)^T$. Let us underline that the sound velocities c_g and c_l chosen for this test are absolutely not realistic. This choice of very low sound velocity values was made in Chanteperdrix, Villedieu, and Vila (2002); Helluy, Golay, Caltagirone, Lubin, Vincent, Drevard, Marcer, Fraunié, Seguin, Grilli, Lesage, Dervieux, and Allain (2005); Golay, Ersoy, Yushchenko, and Sous (2015): it allows to artificially increase the Mach number and to reduce the CFL constraint. In terms of physical interpretation, results have to be carefully considered, especially if compressible effects are indeed non-negligible, which may lead to incorrect results.

The dam break test is simulated on an AMR grid, whose lowest level of refinement is 3 ($\Delta x_{max} = 1.0/2^3$) and highest level is 8 ($\Delta x_{min} = 1.0/2^8$). The refinement criterion is based on $\Delta\alpha$ and the refinement thresholds are $\xi_{\text{refine}} = 1.0 \cdot 10^{-5}$, $\xi_{\text{refine}} = 5.0 \cdot 10^{-6}$.

Computational cost: Given the initial conditions and domain geometry, the flow of liquid reaches the opposite side of the domain within 0.3s and a second wave comes back within 1s. The final time of the computation is 1.5s.

The computation was performed on 64 nodes, of 8 CPU cores each, of the computing Mesocenter of Centrale Paris (Intel Xeon® 5650 processors, QDR Infiniband connectivity). The whole computation took about 4h for 5.59×10^5 time iterations. At the beginning of the simulation, the number of cells was 5.88×10^4 (3.5×10^{-3} compression rate). At the end, due to diffusion and acoustic effects, it reached 1.25×10^6 (7.4×10^{-2} compression rate), while the equivalent uniform mesh would have had around 1.7×10^7 cells.

Visual results: The volume satisfying $\alpha < 0.5$, standing for the liquid phase, as well as the mesh at the domain boundaries are represented in the subfigures of Figure 9.22 at different time steps.

Due to the modified pressure laws, we do not encounter low-Mach difficulties and the solution seems to be physically plausible. However, this observation would require a comparison with experiments or with a reference solution.

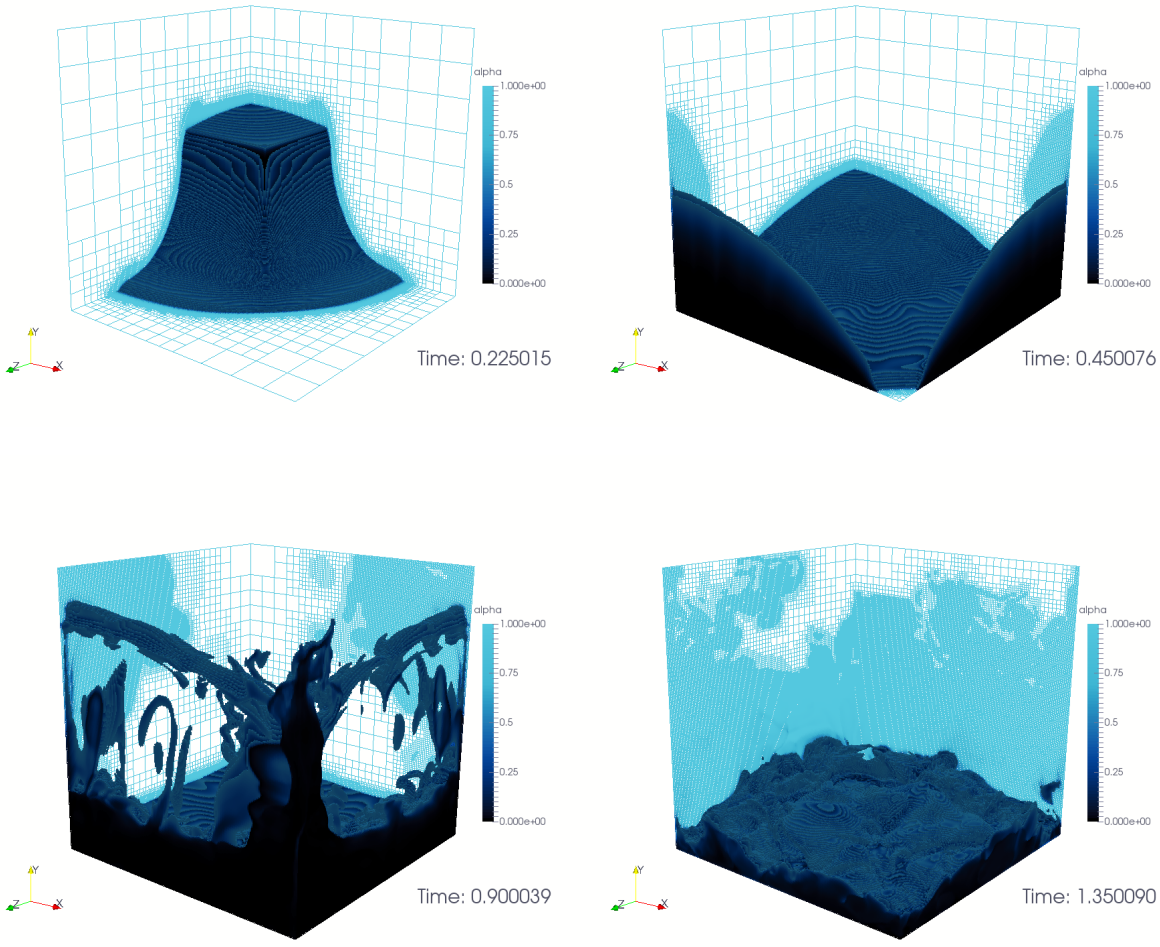


Figure 9.22: *Simulation of a 3D dam break. View of mesh and volume fraction $\alpha < 0.5$. Refinement criterion is* α***-gradient***.

9.7 Conclusion and future works

The aim of this chapter was to verify the numerical schemes in the context of AMR grids and their implementation in CanoP through a series of numerical tests. The order of accuracy of the schemes are preserved on the AMR grids, provided that the refinement criterion and the refinement thresholds are well designed. These parameters have indeed an influence on the compression error of the solution and the compression rate of the grid. In the future, it could be interesting to consider refinement strategies that are closer to the multi-resolution methods, so as to better control both the compression rate and the compression error, as well as the refinement of parts of the solution where small amplitude variations are generated by the numerical schemes.

Another aspect that necessitates further study is the interaction between the solution and the grid in the low-Mach regime for compressible two-phase flows. Although the low-Mach corrections to classic Godunov-type schemes seem to improve the accuracy of the solutions in this regime, several numerical problems still remain to be solved. However, in our tests, these problems were not critical enough to prevent interfacial flows simulations to give results coherent with experimental observations, as is illustrated in section 9.5 with the qualitative comparison of our numerical solution with experimental photographs.

Chapter 10

Dam breaks simulations and study

10.1 Introduction

In Helluy, Golay, Caltagirone, Lubin, Vincent, Drevard, Marcer, Fraunié, Seguin, Grilli, Lesage, Dervieux, and Allain (2005), a benchmark was performed between six models associated with their numerical methods. The test induces the motion of a solitary wave and its breaking when it reaches a submerged reef. Several models and methods were tested in this benchmark, including: two-phase mixture models, incompressible equations, an incompressible approach with a pseudo-compressibility and shallow-water equations. In this paper, the low-Mach problem was not mentioned explicitly although artificial sound velocities were used for the compressible two-phase model.

Following the lines of Helluy, Golay, Caltagirone, Lubin, Vincent, Drevard, Marcer, Fraunié, Seguin, Grilli, Lesage, Dervieux, and Allain (2005), we propose to test the two-phase 3-equation model (9.1). We adopt our low-Mach discretization strategy, by choosing either a triangular grid, or the low-Mach modification of Chapter 6, section 6.5. This modified scheme is based on a HLLC scheme: we will note it HLLC-M and note HLLC the original scheme presented in Chapter 4, section 4.2.3. Results are obtained on AMR grids, with the refinement criterion based on $\tilde{\Delta}\alpha$ (see its definition in Chapter 6, section 6.4). They are compared with the weakly compressible approach of Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) and experimental data from Martin and Moyce (1952); Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014); Stansby, Chegini, and Barnes (1998) for dam break configurations.

The dam break test turns out to be a very instructive test for low-Mach separated phases flows simulation strategies. It has indeed been studied in several works involving different models, numerical methods and experiments.

First in this chapter, we will present an overview of three experimental works from Martin and Moyce (1952); Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014); Stansby, Chegini, and Barnes (1998). This shall lead us to consider three sets of experimental data and two different dam break configurations: the dry-bed and the wet-bed cases. These elements will enable checking:

- the overall flow kinematics, by comparing both height and position of the liquid front at several instants,
- the dynamics effects of the flow with pressure measures and wave breaking process comparisons.

We will present numerical results obtained with the CanoP code, with the approach of Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) and compare both numerical approaches with experimental data.

10.2 Dam breaks experimental configurations

Let us present here the works whose experimental data are considered in this chapter. In the following, a *dry-bed* configuration refers to the configuration for which, initially, there is no water downstream the dam gate ($h = 0$ on figure 10.1). A *wet-bed* configuration refers to a flow with a water layer covering the bed of the domain, i. e. $h > 0$ in Figure 10.1.

10.2.1 Classic sets of measures for dry-bed configurations

The data issued from Martin and Moyce (1952) have been used in numerous papers for comparisons with simulation results or for comparisons with other experimental measures (Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014)). They are indeed considered as references due to the quality of the experimental setup and to the several sets of measures that are available and that ensure a good reproducibility of the data. The main difficulties in such measures lie in the opening of the gate and in taking care of a well-dried surface downstream the dam gate. Indeed, theory and numerical results assume an instantaneous gate opening. Observations show that when the downstream zone is even slightly wet, the dynamics of the dam breaking is very different (Jánosi, Jan, Szabó, and Tél (2004)).

Two quantities are measured by 300 fps cameras and studied in Martin and Moyce (1952):

- the residual height of water $H0(t)$ at the origin of the domain ($X = 0$ on the schematic representation on Figure 10.1),
- the position of the wave surge front $l(t)$, which is recorded from the top by a camera located above the setup.

H and L denote respectively the height and width of the water column at $t = 0$ ($l(0) = L$). Some theoretical aspects are also discussed in Martin and Moyce (1952), so as to get time-evolution models for $H0$ and l. In particular, for rectangular initial shapes, it is shown, using shallow-water theory and Ritter (1892), that the dimensionless quantities $H0/H$ and l/L only depend on the ratio n^2 of H over L: $n^2 = H/L$. The measures of the front velocity in long times agree with the estimate $u_{max} \sim 2\sqrt{gH}$ yielded by the shallow-water theory (Ritter (1892)). At the scale of the experiments, it is assumed that viscosity and surface tension effects are negligible. Another observation from the experiments is that the flow is essentially a 2D problem. However, some 3D effects may be encountered like the interaction of the liquid with the side walls, or the deformation of the front surface, as discussed in Jánosi, Jan, Szabó, and Tél (2004). But as was observed in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014), these deformations appear only for a wave propagating over a long distance. This will not be the case in our simulations.

Different configurations (initial liquid volume shapes) are experienced in Martin and Moyce (1952). We have retained a rectangular initial shape, whose dimensions are given in table 10.2, which corresponds to the ratio $n^2 = 2$.

10.2.2 A more recent contribution

More recently, Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) also performed measures on a dam breaking over dry beds. The aims of their work were:

- to gather precise and reliable data, based on many repetitions of the experiment in a rigorous experimental setup. Particular care was given to well drying the bed before the dam is released,
- to get further information on the pressure loads, especially on the opposite wall, by means of five pressure sensors. Four of them are used to get the pressure loads through time at different heights at the wall (see in table 10.1 P1 to P4 for their exact locations), while the fifth one, P5, is positioned at the same height as P2: it is used to watch at the three-dimensional aspect of the flow.

A detailed description of their experimental setup is given in their paper, where they also discuss the different parameters that could have an influence on the flow dynamics. Error estimations for their

measures are also available. The difference with the work of Martin and Moyce (1952) lies in the statistical data the approach in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) can provide. However, only two configurations are tested in this last paper, while more of these were considered in Martin and Moyce (1952).

Besides pressure measures, the following data are also available in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014):

- the height of the liquid phase is provided at different distances from the dam gate (see on Figure 10.1 and the coordinates of H1 to H4 are given in Table 10.1) with video recordings. These data also include measures when the secondary wave passes. For this period, however, the authors mentioned that the data were not reproducible.
- the front position was recorded by a digital camera.

The flow configuration can be considered as two-dimensional in a first rough approach. Indeed, no deformation of the front surface was observed. However the comparisons of pressure measures from P2 and P5 indicate small variations in the transverse direction.

Data measured in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) are compared with numerous previous works: a detailed discussion about the differences is given in their paper. The authors however noticed that for the early times of the liquid motion, there were differences in the front location with respect to the previous works. They explained this difference either by a different process of the gate removal, or by the fact that the bet might be slightly wet in the previous works. For later moments, the measures are in good agreement with measures of the literature. Similarly, the measures of pressure show certain differences with previous works that are discussed in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014).

Probe name	X location	Y location
H1	0.3	0.0
H2	0.865	0.0
H3	1.114	0.0
H4	1.3625	0.0
P1	0.0	0.003
P2	0.0	0.015
P3	0.0	0.030
P4	0.0	0.080

Table 10.1: *Position of the different probes from Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) experiment.*

10.2.3 Case of wave breaking and wet-bed experiment

In Stansby, Chegini, and Barnes (1998), both dry and wet bed dam breaks are studied, for different heights of the water layer downstream the dam gate. However, as the authors mention in their paper, they could not perform dry-bed configurations over a fully dried bed, due to some leakage through the dam gate. That is why we are only considering here the measures performed for the wet bed configurations.

The dynamics of wet-bed dam breaks differ from the dynamics of dry-bed ones: the first stages of the flow show the formation of a wave that breaks after some time. Then a pocket of air is entrapped within the liquid. This air entrapment phenomenon is interesting to study, since compression effects acting in the gas may lead to damaging effects on solid structures in other and real-sized configurations. The experimental data from Stansby, Chegini, and Barnes (1998), that are associated with a configuration of laboratory size, will allow us to test the ability of the numerical methods to reproduce this wave-breaking phenomenon. Wave breaking phenomenon can also be observed in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014), after the liquid flow has reached the opposite wall and comes back to the dam gate location. The data from Stansby, Chegini, and Barnes (1998) are complementary from these of Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014), since they provide

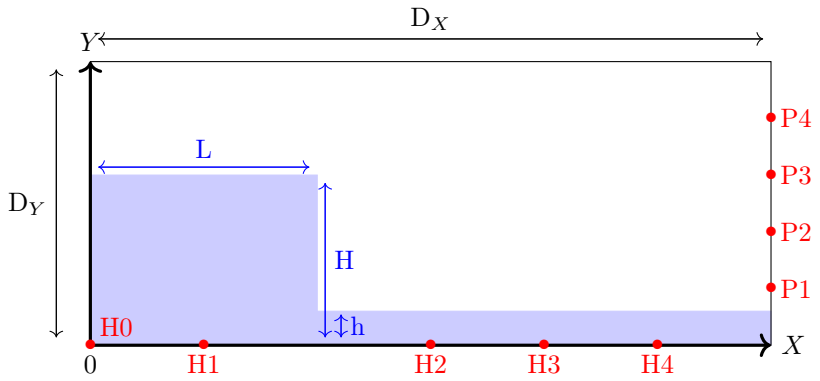


Figure 10.1: *Schematic representation of the dam break configurations.*

information about the shape of the liquid-gas interface, which are not available in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014).

The experimental data were acquired by recording the flow with a video camera that followed the front position. Perspex powder spread at the surface of the liquid volume before the gate was opened, and a laser light sheet helped better visualizing the free surface through time. Among the different configurations experienced, we have chosen one for our simulations which is detailed in Table 10.2.

Finally, let us mention that a model is also proposed in Stansby, Chegini, and Barnes (1998) that allows to derive an analytical pressure profile for the early instants of the dam break problem.

	Martin and Moyce's configuration	Lobovsky's configuration	Stansby's configuration
L	0.3	0.6	0.6
H	0.6	0.3	0.1
h	0	0	0.01
D$_X$	1.0	1.61	1.2
D$_Y$	2.0	0.8	0.2

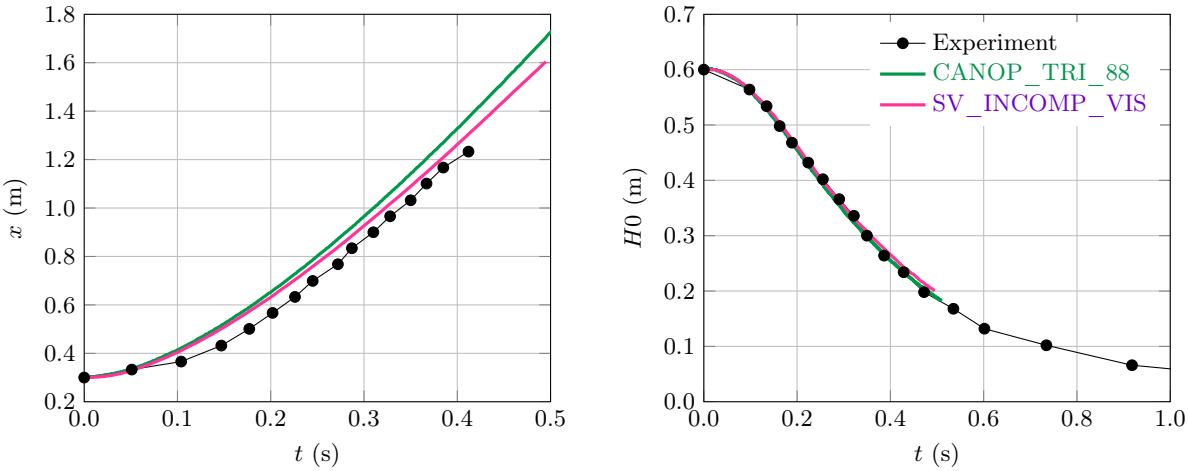
Table 10.2: *Dimensions of the different configurations tested in this study and positions of the probes for Lobovsky's experiment.*

10.3 Comparisons of the numerical and experimental solutions

We will now compare the dam break experimental data presented in the previous sections to several numerical methods and models. We will consider simulations performed with the scheme of Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) for a model made of two viscous incompressible fluids, separated by a sharp interface. The interface is captured through a VOF method. For more details about the numerical method, we refer the reader to Vincent, Caltagirone, Lubin, and Randrianarivelo (2004); Vincent, Larocque, Caltagirone, and Pianet (2007); Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010); Vincent, Sarthou, Caltagirone, Sonilhac, Février, Mignot, and Pianet (2011). Let us mention that the approach proposes also a weakly compressible variant, the results of which are presented here only for the wet-bed configurations, but are part of a more thorough work in collaboration with S. Vincent (Drui, Kokh, Larat, Massot, and Vincent (2017)).

The results obtained for the incompressible model are computed on uniform Cartesian grids. The results obtained with the CanoP code are computed either on the same uniform grid, or on AMR grids whose characteristics will be given for each simulation. Moreover, three discretization strategies are considered:

- the HLLC scheme on a quadrangular mesh,
- the HLLC scheme on a triangular mesh,



(a) Position of the surge front through time.

(b) Residual height of water.

Figure 10.2: *Comparison between data from Martin and Moyce (1952) and a simulation using CanoPand uniform triangles of levels 8.*

- the HLLC-M scheme (with the all-regime modification) on a quadrangular mesh.

In the sequel, we will use the following nomenclature to name the results obtained by the different approaches:

- the simulation results from Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) code with the incompressible fluids will be noted SV_INCOMP_VIS ,
- the simulation results from Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) code with weakly compressible fluids will be noted SV_COMP_VIS ,
- the simulations from the compressible two-phase model on AMR grids will be noted:
  - CANOP_TRI_mM for a triangular AMR grid of minimum refinement level m and maximum M,
  - CANOP_QUAD_mM for a quadrangular AMR grid of minimum refinement level m and maximum M,
  - CANOP_LM_mM for a simulation using the HLLC scheme with the low-Mach correction, over a quadrangular AMR grid of minimum refinement level m and maximum M.

For the AMR simulations, the refinement criterion is based on the volume fraction gradient. The refinement thresholds are $\xi_{\text{refine}} = 1.0 \cdot 10^{-5}$ and $\xi_{\text{coarsen}} = 5.0 \cdot 10^{-6}$ (we refer to their definition in Chapter 6). Moreover, as the interface is diffused by the numerical method, the interface is chosen to be situated at isoline $\alpha = 0.5$.

10.3.1 Comparison for Martin and Moyce's experiment:

The configuration studied in this test is given in Table 10.2. The results for the different simulations and comparisons for the position of the surge front and the residual height with Martin and Moyce (1952) data are given in Figure 10.2. One can note that, in the computation using CANOP_TRI_88 , the simulated front is in advance with respect to the experimental and SV_INCOMP_VIS ones, while the residual height of water is in good agreement with the experimental measures. The lack of viscosity and of boundary friction may explain that the liquid phase flows too fast.

10.3.2 Comparison for Lobovsky's experiment:

The configuration studied in this test is given in Table 10.2. The data of Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) complete the other experiment studied here, since they provide information about the flow until it comes back leftward, after the wave has reached the opposite wall (see illustrations of the flow at different time of the simulation in Figure 10.3).

Figure 10.4 and Figure 10.5 display respectively comparisons data between numerical results and experiments for the front position and the liquid heights at locations H1, H2, H3 and H4 (from Table 10.2).

Until the initial wave formed by the motion of the water column reaches the right boundary of the domain, we can make similar observations as in section 10.3.1:

- the numerical front is slightly ahead of the experimental data,
- the residual height H1 (Figure 10.5a) is in good agreement with the experiment.

On the other hand, for the liquid height acquired at positions H2 to H4 (Figure 10.5b to Figure 10.5d), the numerical results show a good match with the measures, except in one aspect: one can observe a small hump in the measures at the beginning of the signal (near time $t \cdot \sqrt{g/H} \approx 2$) for H3 and H4. This hump is partially reproduced with SV_INCOMP_VIS simulation, but it does not appear on CANOP_TRI_18 simulation.

Numerical and experimental solutions present more significant differences when the liquid wave travels back leftward. The best agreement is for the location H4. For H3, the numerical solution is slightly in advance, but does not predict a height as high as the experimental data. This is the contrary at H2, where numerical and experimental signals are synchronized, but the liquid height of CANOP_TRI_18 remains higher than the experimental measure. Finally, for H1, the numerical solution is clearly ahead of the experimental one. As noticed in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014), due to the mixing zone that is created when the wave breaks, these height measures are not repeatable. It is then difficult to draw here a conclusion about the differences between the numerical and the experimental solutions.

Let us now examine the pressure data. One can note that, due to the overestimated velocity of the liquid front by CANOP_TRI_18 simulations, the start of pressure increase appears sooner with CANOP_TRI_18 result than in the measures (see Figure 10.6a) and than in SV_INCOMP_VIS simulations (see Figure 10.6b). SV_INCOMP_VIS computations give moreover results in very good agreement with the experimental measures, except for the lowest probe P1. The magnitude of the pressure signal in CANOP_TRI_18 simulation is quite far from the experiments for P1 (see Figure 10.6a). However, as was mentioned in Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) about the flow dynamics occurring near P1, it is the probe that gets the higher load from the liquid wave. The pressure measured at probe P4 is also clearly underestimated by CANOP_TRI_18 solution. Moreover, the quick decrease of the load that is observed experimentally after the pressure peaks is not well represented by CANOP_TRI_18 for P1 to P3 probes: the pressure decrease of the numerical solution is indeed slower in Figure 10.6a. Finally, one can note that when the wave breaks, a quite smooth pressure bump is measured. This pressure bump is also observable numerically. Through a grid convergence analysis of paragraph 10.3.4, we will investigate the differences between CANOP_TRI_18 and the experiment.

10.3.3 Comparison for Stansby's configuration and wave breaking

We consider here a wet-bed case with initial data described in Table 10.2. We reproduce here the experiment of Stansby, Chegini, and Barnes (1998).

The shape of the interface is given at two different times in Figure 10.7. One can make the following observations:

- the front of the wave predicted by the numerical solutions is in good agreement with the experimental measures: the wave propagation seems to be correctly captured,

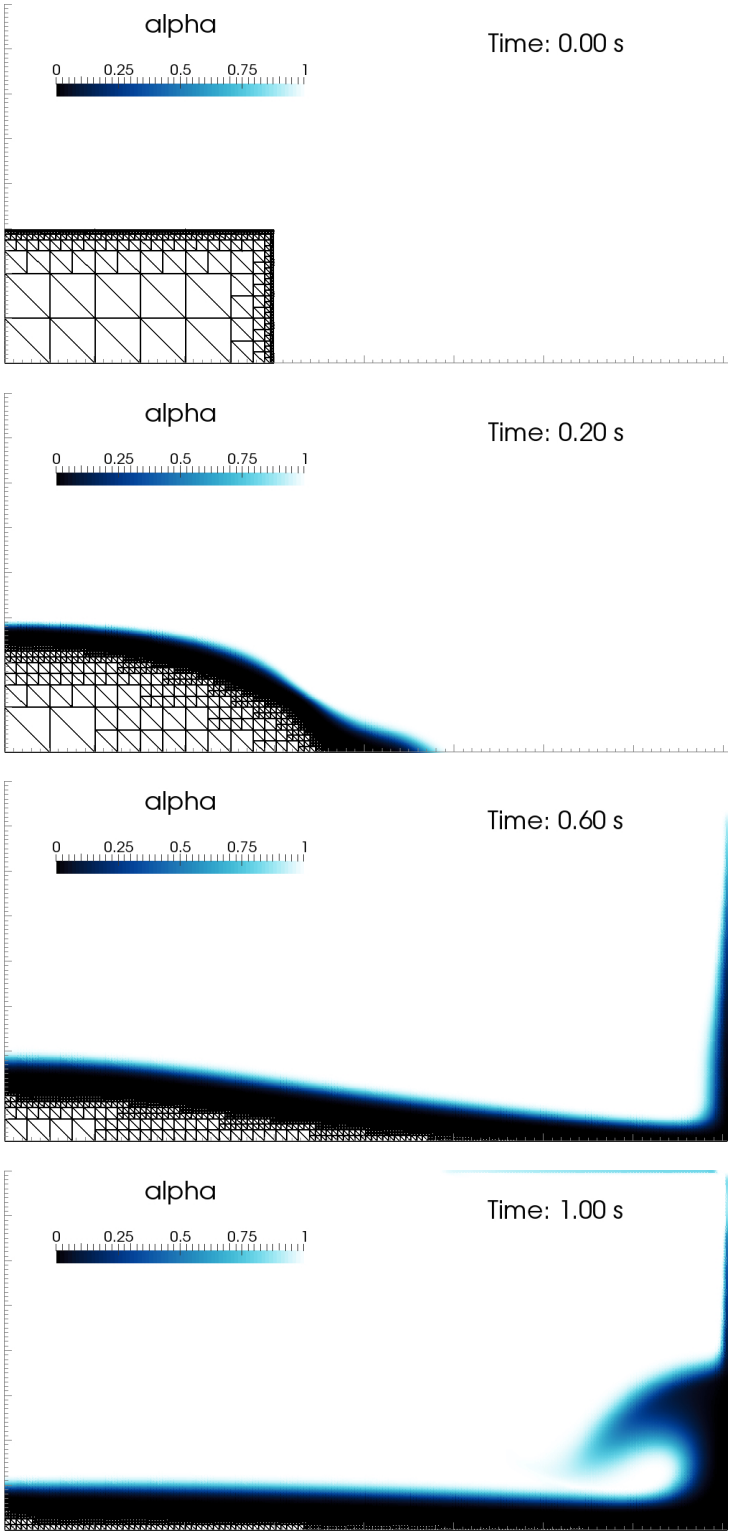


Figure 10.3: *Simulation of the dam break with Lobovsky's configuration. Numerical results (volume fraction α values represented on the adaptive grid) obtained with CANOP_TRI_18 at different times of the simulation.*

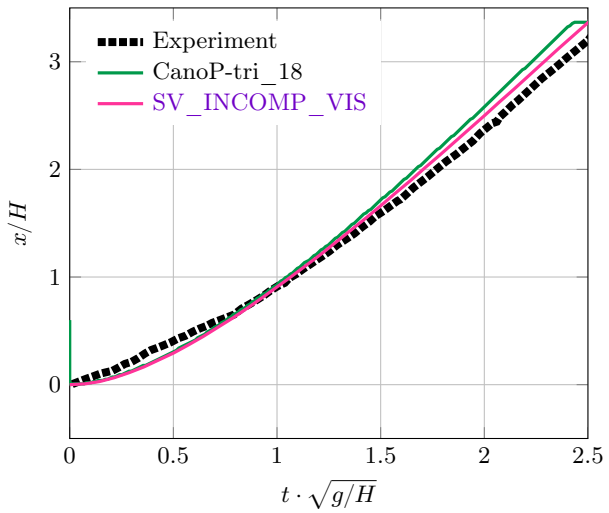


Figure 10.4: *Comparison of the surge front position between data from Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014), an AMR simulation of levels 1-8 with triangles (*$512 \times 256 \times 2$ *equivalent uniform mesh) and SV_INCOMP_VIS .*

- for the three numerical solutions, the height of the wave is of the same order, and is slightly lower than the experiment data. This could be explained by the mixing zone created by the breaking wave, that makes the tracking of the interface quite complex in the experiment,
- air is entrapped by the liquid wave. While there is no experimental data about this air pocket, both SV_INCOMP_VIS and SV_COMP_VIS give the same results. On the other hand, the numerical solution given by CANOP_TRI_88 is very diffused in this zone, and the multiple air pockets found in SV_INCOMP_VIS and SV_COMP_VIS are not visible with CANOP_TRI_88 .

Further investigation may for example rely on the work of Jánosi, Jan, Szabó, and Tél (2004) that could bring more comparison elements concerning the wave breaking and the air pockets entrapment.

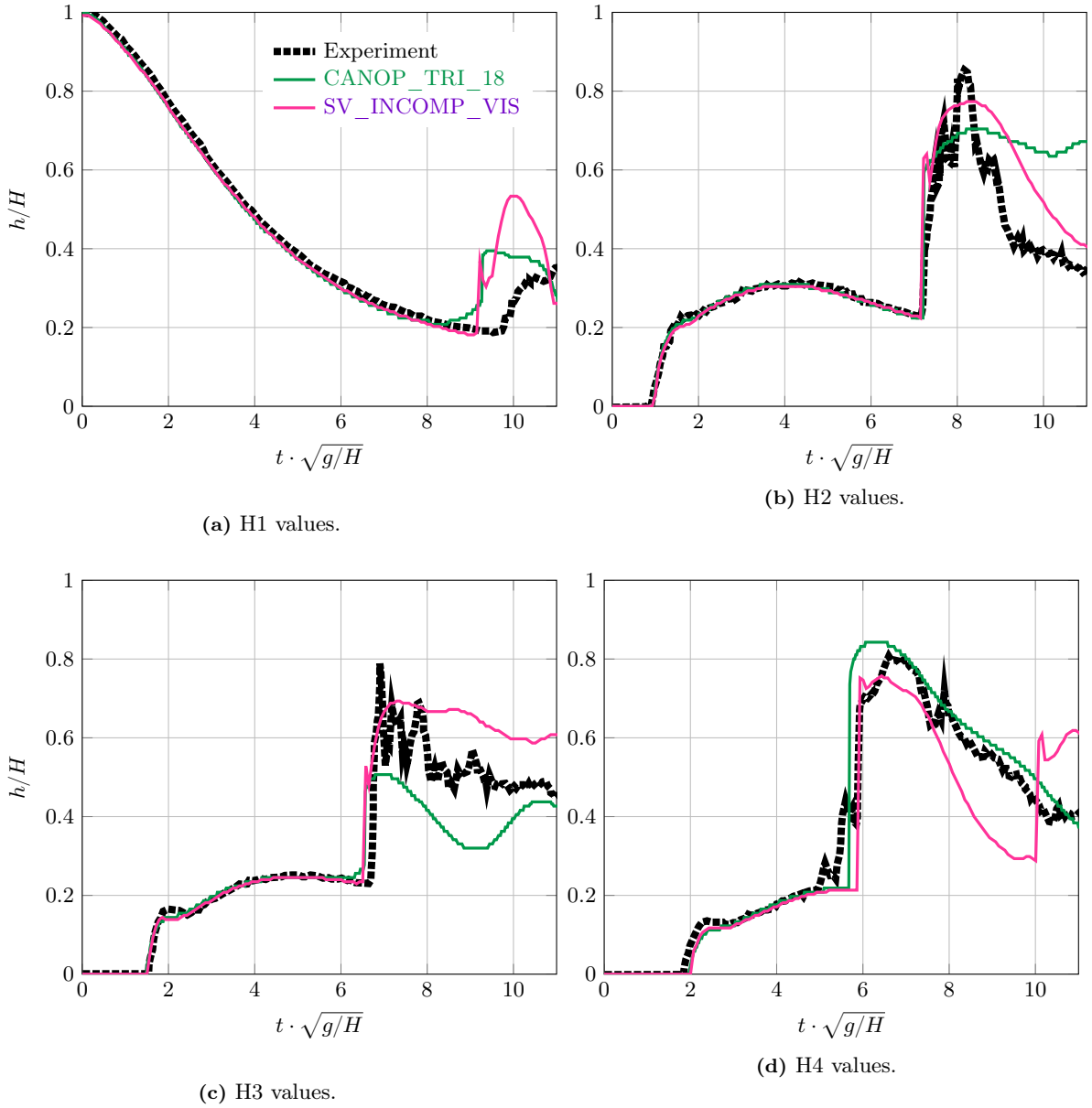
10.3.4 Convergence results for the two-phase compressible approach

As we have noticed in the previous paragraphs, comparisons of the pressure obtained numerically at probes P1 to P4 differ from the experiment: the pressure overload, due to the wave, begins sooner in the numerical solution than in the experimental one, while the load peak is lower. Moreover, the wave seems to move faster in the numerical solution. Here, we are then studying the convergence of the simulations CANOP_TRI_mM when the mesh is refined, *i.e.* when the maximum level of refinement M is increased. In particular, we wonder whether the simulations can better fit the experimental data, with respect to the motion of the surge front, the liquid heights and the pressure loads. With a series of different grids, we draw here the main trends of numerical solutions and propose an analysis of the differences between numerical results and experiment.

We present in Figure 10.8 and Figure 10.9 comparisons of liquid heights and pressure loads for three different triangular grids:

- a uniform grid with cells of levels 8 that corresponds to $\Delta x = 3.125 \cdot 10^{-3}$m,
- a uniform grid with cells of levels 9 that corresponds to $\Delta x = 1.5625 \cdot 10^{-3}$m,
- an adapted grid with lowest level of refinement of 8 that corresponds to $\Delta x_{max} = 3.125 \cdot 10^{-3}$m and highest level of 10, *i.e.* $\Delta x_{min} = 7.8125 \cdot 10^{-4}$m.

One can see on these figures that the results for the liquid heights are not changing significantly with the mesh resolution: in particular, for the probe H4, the first little hump is still not captured numerically and the wave is still coming ahead of the experiment.



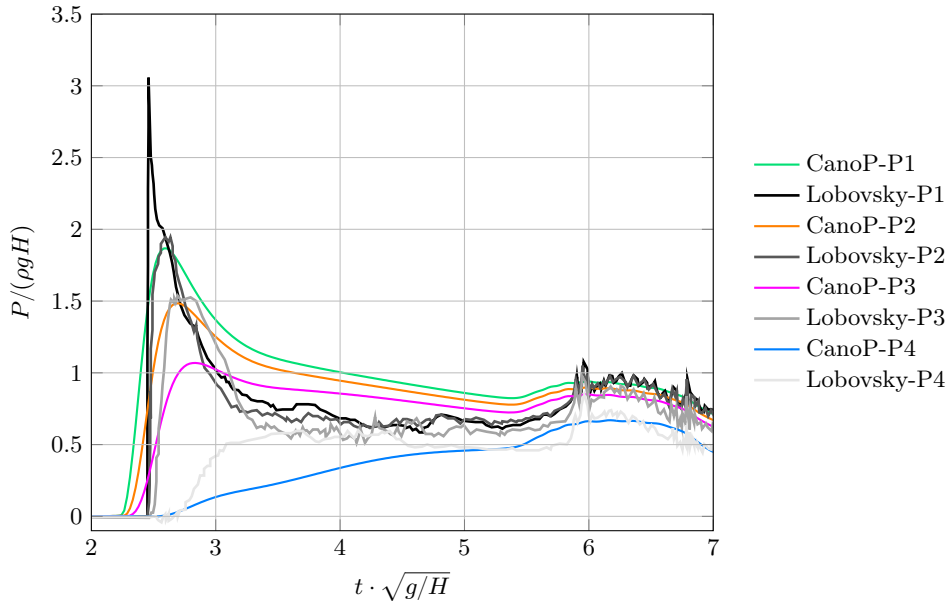
(a) H1 values.

(b) H2 values.

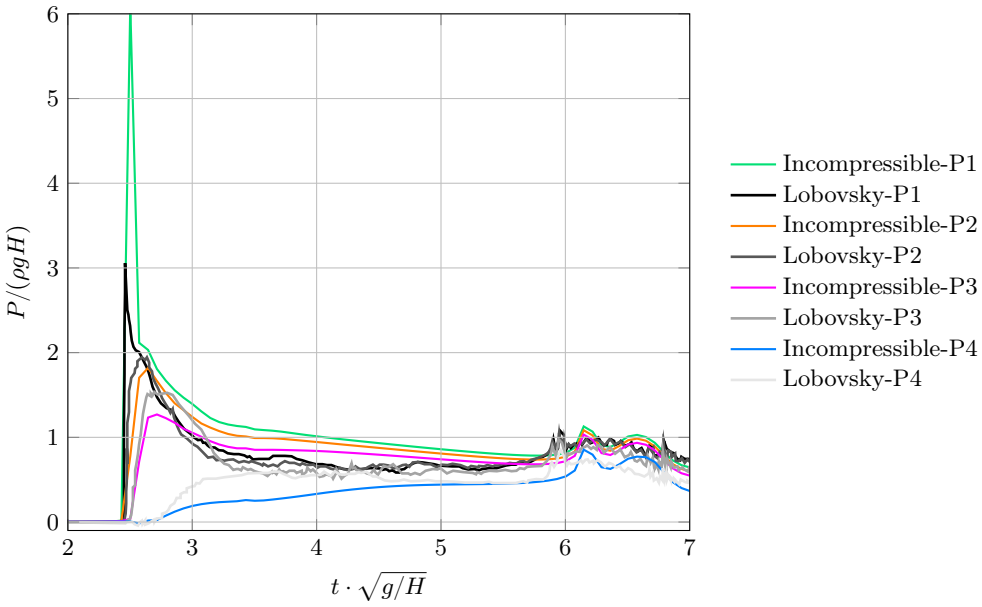
(c) H3 values.

(d) H4 values.

Figure 10.5: *First comparison between data from Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014), CANOP_TRI_18 and SV_INCOMP_VIS . Liquid heights at the different probes through time.*



(a) Comparison between experimental data (grey curves) and CANOP_TRI_18 .



(b) First comparison between experimental data (grey curves) and SV_INCOMP_VIS .

Figure 10.6: *Experimental pressure values at the different pressure probes from Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) and comparisons with the numerical results.*

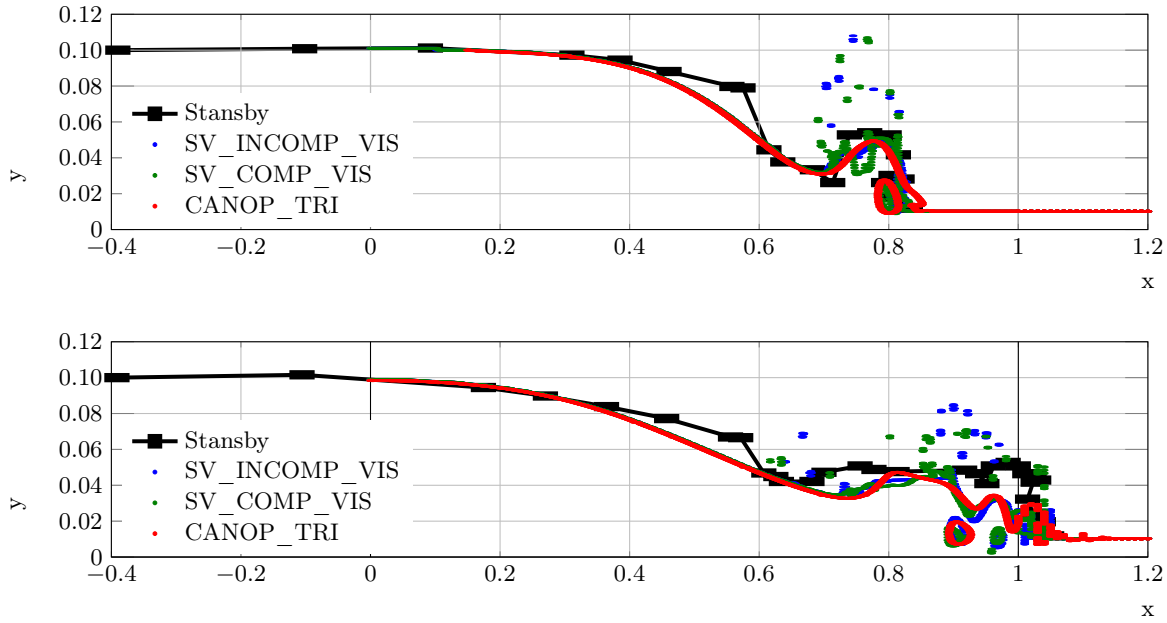


Figure 10.7: *First comparison between SV_COMP_VIS in green, SV_INCOMP_VIS in blue, CANOP_TRI_88 in red and Stansby's measures in black at times $t = 0.24s$ (upper graph) and $t = 0.4s$ (lower graph).*

Concerning the pressure measures (Figure 10.9), we can observe that:

1. the pressure peaks for all probes are better resolved when the mesh is finer: their values are higher when the grid is refined,
2. the beginning of the signal of pressure overloads is more delayed as the mesh gets finer. This is particularly visible for pressure probes P1 to P3.

The first point may be explained by the decrease of numerical dissipation with finer grids. Indeed, this numerical dissipation acts as a viscosity. Thus, the higher it is, the higher the fluid momentum gets dissipated.

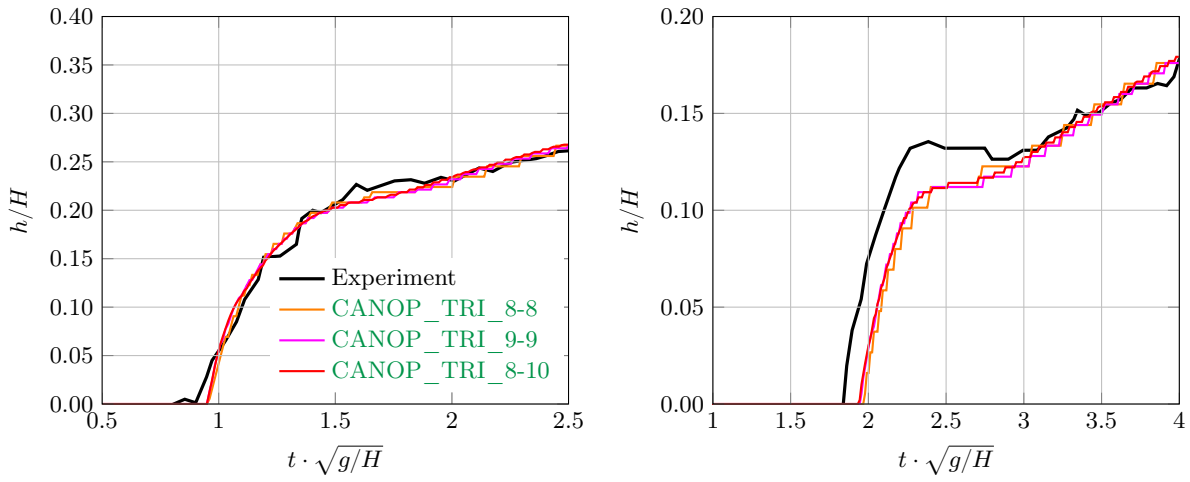
The second point is another consequence of the numerical dissipation decrease: with a higher resolution, the diffusion of the interface is reduced. However, this diffusion is characterized by the spreading over few cells of a numerical gas/liquid mixture, whose density lies between the density of the gas and the liquid. Then, this diffusion layer reaches the probes before the real liquid wave, and generates pressure loads, that are not necessarily physical. The reduction of the diffusion layer, enables to delay the beginning of this pressure loads.

10.3.5 Conclusion of the comparisons with experiments:

With the study presented in this first part, we have compared the numerical solutions obtained with the CanoP code, using HLLC scheme on triangular AMR grids, with three sets of experimental data and with the scheme of Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) for incompressible or weakly compressible fluids.

The first data, issued from Martin and Moyce (1952) are considered as standard and used in many works. In the same configurations as the experiment, our numerical solutions show good agreements with the experimental data:

- although the front of the head is slightly ahead in the numerical results, the time gap in constant after some time, meaning that the propagation velocity is of the same order of magnitude,
- the height of the residual water column is similar in numerical and experimental results.



(a) H2 values

(b) H4 values

Figure 10.8: *A convergence study for the simulations using CanoPover triangular meshes (levels 8-8 in orange, levels 9-9 in pink, levels 8-10 in red). Representation of the liquid heights recordings at two different probes.*

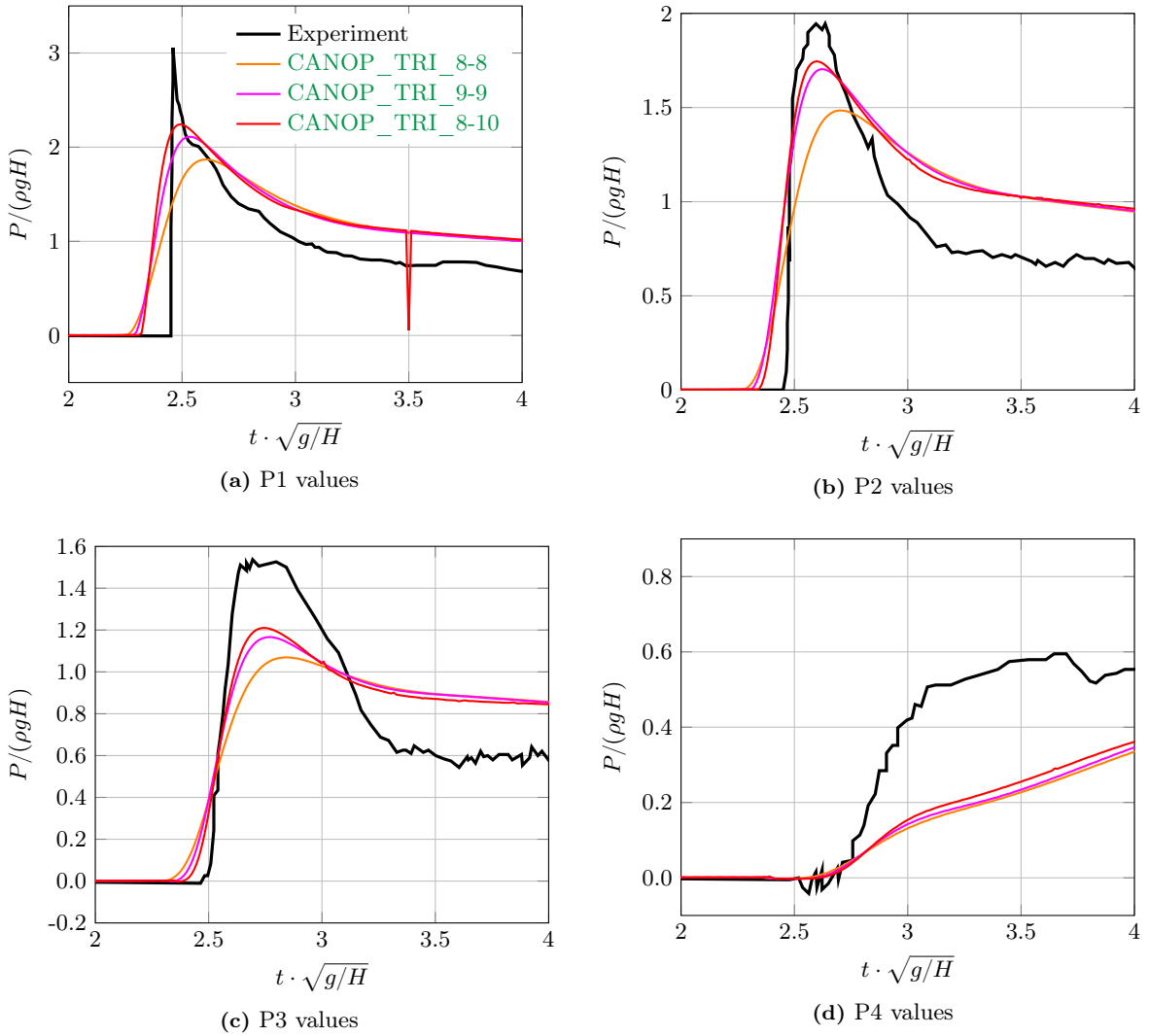
The second data come from the work of Lobovsky, Botia-Vera, Castellana, Mas-Soler, and Souto-Iglesias (2014) and are complementary to the data of Martin and Moyce (1952), since the height of water is measured at more locations and for longer times, and pressure data are available. When comparing our numerical results with the experimental ones, on can draw the same observations as with the comparison with Martin and Moyce (1952), up to the moment the wave has reached the opposite wall and comes back to its original place. There, some delays can be observed between numerical and experimental results. Moreover, some differences in the pressure signals are observed:

- the pressure signal starts sooner in the numerical results than in the experimental ones,
- the pressure peaks at the different probes are lower in the numerical results than in the experimental ones.

However, these differences seem to be reduced when the discretization of the numerical solution is refined (with finer grids): the pressure peaks are higher, and the pressure signal provided by the numerical solution is closer to the experimental one.

Finally, the numerical settings have been tested on a wet-bed configuration, and the shape of the flow was compared with the data of Stansby, Chegini, and Barnes (1998). Both solutions from CanoP and from the approach of Pianet, Vincent, Leboi, Caltagirone, and Anderhuber (2010) are in good agreement with the experiment. However, this experiment provides few information. It would be interesting to consider other comparisons for this test case.

In all this part, we have considered only computations using HLLC scheme on triangular grids. The pressure instabilities that were mentioned in Chapter 9 for the case of the bubble have also been observed here, but they did not prevent the numerical solutions to match the experimental ones. Let us now study in more details the HLLC scheme computed on quadrangular grids, as well as the HLLC-M scheme.



(a) P1 values

(b) P2 values

(c) P3 values

(d) P4 values

Figure 10.9: *A convergence study for the simulations using CanoPover triangular meshes (levels 8-8 in orange, levels 9-9 in pink, levels 8-10 in red). Representation of the pressures recordings at the different pressure probes.*

10.4 An analysis of the low-Mach problem in dam breaks configurations

10.4.1 Comparison of the shape of the interface:

To better understand the issue related to the low-Mach configuration, let us first have a look at a comparison between different numerical solutions for a dry-bed configuration (here, Lobovsky's configuration of table 10.2). In Figure 10.10, we present the results for the liquid surface shape. One can see that the results given by CANOP_QUAD_88 does not match the other results, that are CANOP_TRI_88 , CANOP_TRI_18 , CANOP_LM_88 , CANOP_LM_18 and SV_INCOMP_VIS . The liquid motion is much slower with CANOP_QUAD_88 than with the other numerical solutions. This defect is even accentuated with CANOP_QUAD_18 , *i.e.* when using an AMR quadrangular grid: there, only the foot of the dam seems to move, while the rest of the liquid column is still.

When using the AMR grid, we could discuss the choice of $\tilde{\Delta}\alpha$ as the refinement criterion: maybe should it be preferable to use also a criterion based on pressure and velocity variations? We have seen in the comparisons with the experimental data of the previous sections, that the choice of the variations of α as the refinement criterion seems to be correct with HLLC on a triangular grid. Moreover, the same AMR cartesian grids with the all-regime HLLC-M scheme gives also good results, as one can see on Figure 10.10.

In the sequel, we choose to study the low-Mach problem in the first instants of the dam-break motion, *i.e.* where a particular pressure profile is established (as was studied in Stansby, Chegini, and Barnes (1998)).

10.4.2 Starting up of the dynamics

In the real physical configuration, both gas and liquid stand at hydrostatic equilibrium at each side of the dam gate, before this gate is removed. Consequently, due to the difference in density between the two fluids, the hydrostatic pressure gradient is larger in the liquid than in the gas. Once the gate *disappears* (which is not possible in reality, but one may suppose that it is removed quickly enough in the experiments so as to have little influence on the flow dynamics), a disequilibrium then exists at the vertical interface between gas and liquid. From this moment, the only possible equilibrium state is obtained when the liquid phase horizontally lies at the bottom of the domain.

The whole motion of the liquid phase can be divided into different stages:

1. the initial disequilibrium modifies the pressure profile inside the liquid phase (see Stansby, Chegini, and Barnes (1998) and 10.13),
2. liquid momentum is then increased and at the bottom of the liquid column, where the pressure gradients are the strongest, liquid begins to move to the right,
3. the entire liquid column then flows towards the opposite side of the domain.

Let us study the numerical results obtained during the first two stages. These stages present indeed the characteristics of being at a very low Mach number regime (as $M = 0$ initially).

10.4.2.1 Different initial states and first acoustic effects

In a numerical point of view, one has to define the initial condition for the dam break problem. It is given at $\mathbf{x} = (x, y)$ by:

$$\begin{cases} p(0,\mathbf{x}) &= f(\mathbf{x}), \\ \alpha(0,\mathbf{x}) &= g(\mathbf{x}), \\ \mathbf{u}(0,\mathbf{x}) &= \mathbf{0}, \end{cases} \tag{10.1}$$

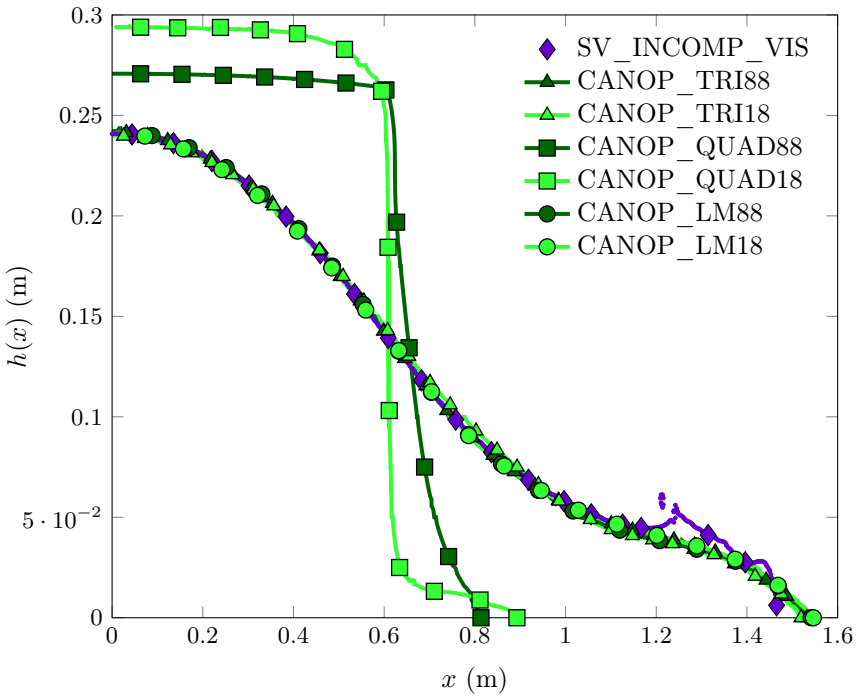


Figure 10.10: *Comparison of the profiles for iso-contours $\alpha = 0.5$. Space step is $\Delta x = 3.125 \cdot 10^{-3}$ m for each simulation. Purple marks are for SV_INCOMP_VIS solution. Green triangles are for triangular meshes on CanoP(dark ones for uniform mesh, light ones for triangular AMR). Green squares are for quadrangular meshes on CanoP(dark ones for uniform mesh, light ones for AMR). Green circles are for quadrangular meshes and corrected HLLC flux on CanoP(dark ones for uniform mesh, light ones for AMR).*

with:

$$g(\mathbf{x}) = \begin{cases} 0.0 & \text{if} \quad y < H \text{ and } x < L, \\ 1.0 & \text{else.} \end{cases}$$

Now, the two following initial pressure profiles can be considered:

- for a sake of simplicity, we could impose a uniform initial pressure:

$$f(\mathbf{x}) = p^o,$$

- we can define a pressure profile closer to the hydrostatic equilibrium by:

$$f(\mathbf{x}) = p^o + \begin{cases} c_l^2 \rho_l^o \left(\exp\left(-\frac{g(y-H)}{c_l^2} \right) - 1 \right) & \text{if} \quad g(\mathbf{x}) = 0.0, \\ c_g^2 \rho_g^o \left(\exp\left(-\frac{g(y-H)}{c_g^2} \right) - 1 \right) & \text{else,} \end{cases}$$

  where $c_g = 3.4$, $\rho_g^o = 1.0$, $c_l = 15.0$, $\rho_l^o = 1.0 \cdot 10^3$ $p^o = 1.0 \cdot 10^5$ and $g = 9.81$.

Now, let us discuss and compare the results for these two initial configurations.

Uniform initial condition: with an initial condition made of a uniform pressure profile, in the first time iteration with CanoP, all vertical (and horizontal) numerical fluxes are null or balanced. The integration of the gravity source term only generates vertical momentum, that drives the fluids downwards. At the next time steps, due to the reflective boundary condition at the bottom of the domain, momentum starts to accumulate in the lowest ranges of cells, leading to an increase of pressure in this zone. This accumulation propagates gradually upwards, like a compression wave, and finally forms an hydrostatic profile. This description corresponds to the observations made on the numerical solution, and that is illustrated in Figure 10.11.

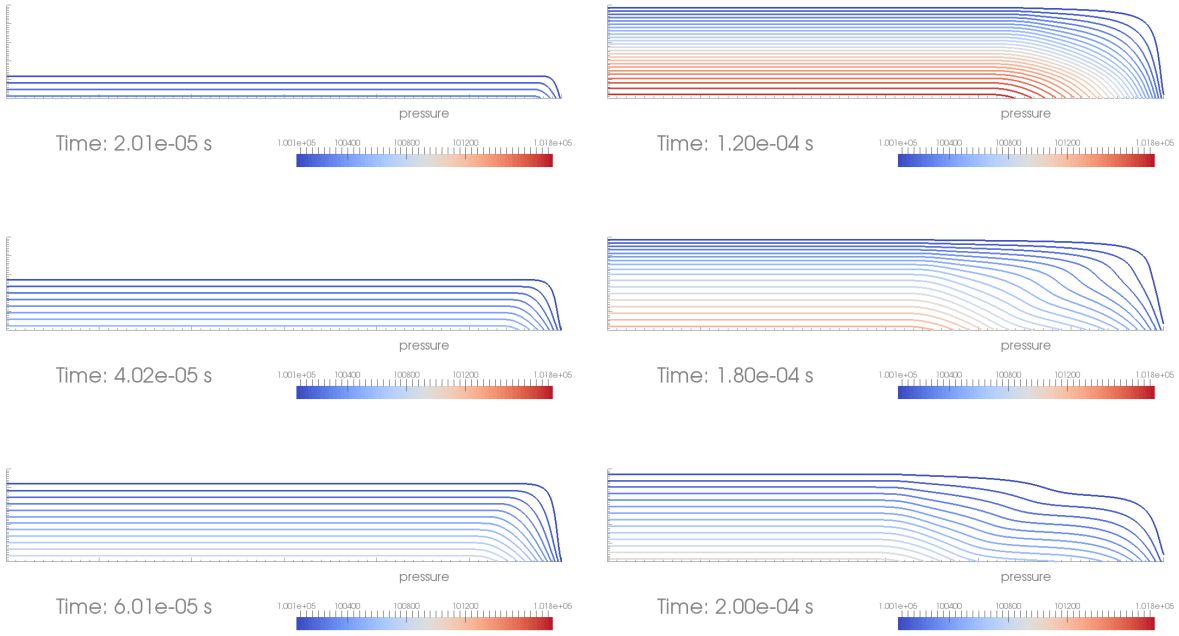


Figure 10.11: *Setting up of the pressure inside the liquid phase for the compressible schemes and a uniform profile in pressure initial state (here CANOP_TRI_88 solution, but identical results with SV_COMP_VIS).*

One can also note that, once the compression wave has reached the upper liquid-gas interface, then, due to the difference of acoustic impedance between the two fluids, one part of the wave is reflected (vertically and horizontally), and one part is transmitted in the gas. These waves propagate and reflect at the domain borders, as well as in the fluids interface until it is entirely dissipated by the numerical diffusion. This wave can be seen as a numerical artifact and may be explained on a one hand by the non-physical initial condition, on the other hand by the non well-balanced character of our numerical scheme (the gravity source term and the fluxes do not exactly balance over a time step).

Hydrostatic equilibrium initial condition: with an hydrostatic equilibrium profile initially set in each phase, the compression waves that propagate vertically are not as much visible as in the previously described configuration. They nevertheless exist since the numerical scheme is not *well-balanced*.

On the contrary, we can observe in Figure 10.12 an expansion wave in the liquid phase (and a compression wave in the gas) due to the initial difference of pressure between liquid and gas at the vertical interface. This wave propagates mainly horizontally in the compressible fluids. It reflects successively and several times at the vertical boundaries of the domain, and at the vertical gas-liquid interface. The wave is finally dissipated by the numerical dissipation.

Despite these pressure waves, we can see in Figure 10.12 that the hydrostatic pressure profile is maintained in the zones of the flow far from the gas-liquid interface, while the pressure profile that leads to the liquid motion develops near the vertical interface.

So, what is the influence of the initial condition for the numerical solution of the dam break problem? For the tests that were used in the comparisons with the experimental measures in the previous sections, we did not notice any significant difference between the two options. However, a quantitative estimation of the characteristic times for the setting of the pressure profiles and the beginning of the liquid sloshing would probably be necessary for real-sized configurations.

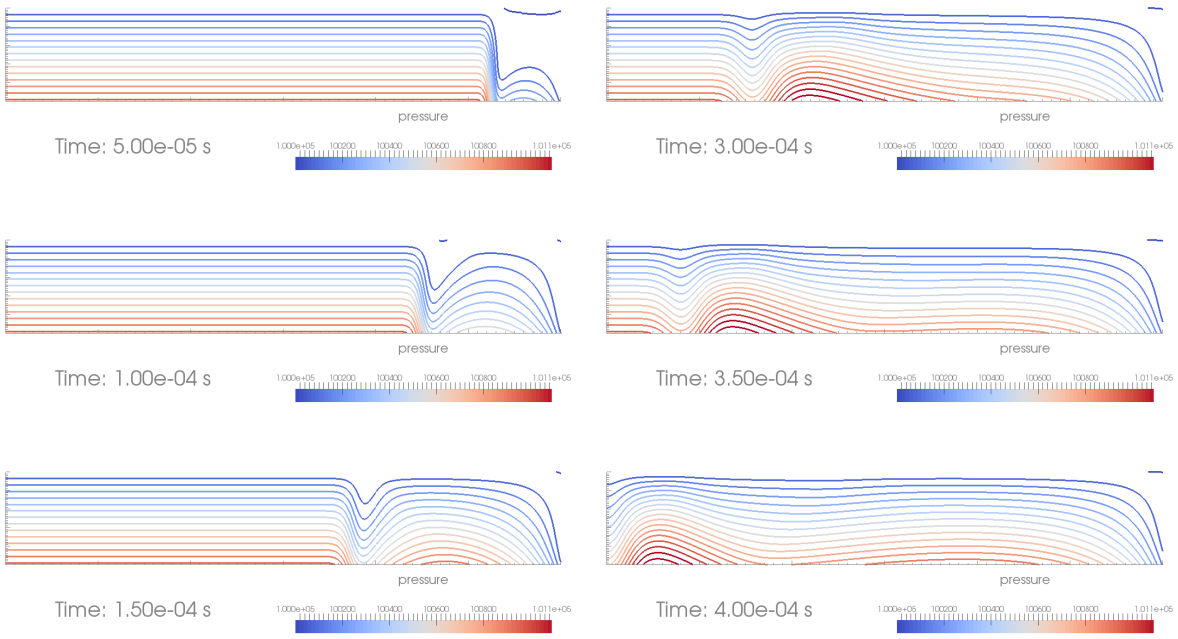


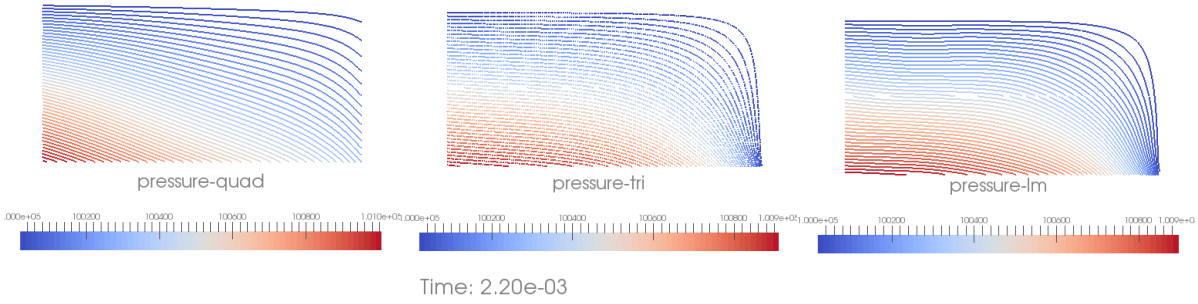
Figure 10.12: *Setting up of the pressure inside the liquid phase for the compressible schemes and a hydrostatic profile of pressure initial state (here CANOP_TRI_88 solution, but identical results with SV_COMP_VIS).*

10.4.2.2 Pressure and velocity profiles before sloshing dynamics

Now let us study the pressure and velocity field profiles developed during these first instants of the motion for the HLLC scheme computed on a quadrangular uniform grid (CANOP_QUAD_88), the HLLC scheme computed on a triangular uniform grid (CANOP_TRI_88) and the all-regime HLLC-M scheme computed on a quadrangular uniform grid (CANOP_LM_88).

One can see on Figure 10.13, that CANOP_TRI_88 and CANOP_LM_88 solutions have similar pressure profiles (which look like the incompressible solution found in Stansby, Chegini, and Barnes (1998)). However, the solution given by CANOP_QUAD_88 is different: the pressure isolines do not join at the foot of the liquid column, like in CANOP_TRI_88 and CANOP_LM_88 . The pressure gradients are not oriented in the same directions and the horizontal pressure gradient at the origin of the liquid motion is underestimated in CANOP_QUAD_88 . Thus, as one can not in Figure 10.13, the velocity field of CANOP_QUAD_88 is made of lower values than the velocity fields in CANOP_TRI_88 and CANOP_LM_88 . This seems to explain then the significant difference of the shape of the interface in Figure 10.10.

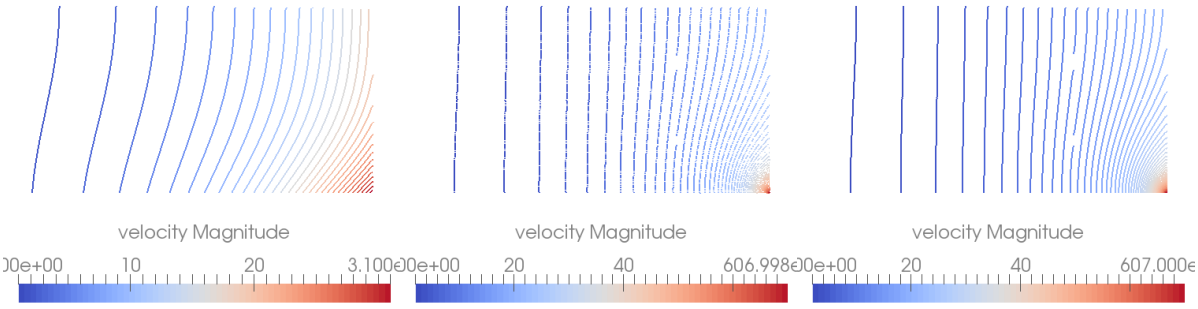
A further study using Hodge's decomposition of the flow solution during these first moments, could bring some answers on the bad behavior of the compressible model, but this is left for a future study.



(a) Pressure profile obtained by CanoPwithout any Low-Mach correction.

(b) Pressure profile obtained by CanoPand triangles.

(c) Pressure profile obtained by CanoPwith the Low-Mach correction.



(d) Velocity magnitude profile obtained by CanoPwithout any Low-Mach correction.

(e) Velocity magnitude profile obtained by CanoPand triangles.

(f) Velocity magnitude profile obtained by CanoPwith the Low-Mach correction.

Figure 10.13: *Setting up of the pressure and velocity inside the liquid phase and at the vertical interface for the compressible schemes just before sloshing (here with CanoP, but identical results with SV_ COMP_ VIS).*

10.5 Conclusion

This preliminary study on dam break test cases has allowed to verify that our numerical strategy, based on the 3-equation compressible two-phase model, the all-regime finite volume schemes and the resolution on AMR grids, is well-suited for separate-phase configurations, even in a low-Mach regime, when one uses the triangular grid or the all-regime scheme. The numerical results have indeed been compared with experimental data, and they show a physical coherence. The numerical dissipation of the scheme seems to explain the slight differences, since a finer discretization improves the numerical results. Finer analyses are under study and will be presented in Drui, Kokh, Larat, Massot, and Vincent (2017). In particular, we will study the compression effects during the wave breaking process. These compression effects may be important to predict numerically, for example in simulations of fluid-structure interactions (see a discussion of the impact of waves on coastal infrastructures in Hayatdavoodi and Ertekin (2016)). With the incompressible equations, such phenomena cannot be reproduced.

Moreover, we better understand the influence of the low-Mach issue in the dam break problem. The poor results of classic Godunov-type schemes solved on quadrangular grids come from the initial instants of the motion, when the pressure and velocity profiles are formed: without a *low-Mach treatment*, these profiles are different from the profiles computed for incompressible flows, and prevent the liquid from flowing. Further investigations on the low-Mach issues could be considered by means of an analysis of the Hodge-Helmholtz decomposition.

General conclusion and perspectives

Conclusions

The injection of liquid fuel in combustion chambers and in particular in Diesel engines, is one important factor with respect to the quality of the combustion and the formation of pollutants. To help designing new devices or to better understand the combustion mechanisms, the numerical simulation is today a widely developed approach, due to the increasing performance of the computational resources. However, the injection of liquid fuel and its burning are multiphase and multi-scale processes, where all physical scales play an important role: from turbulent structures, to the deformation of the gas-liquid interface due to instabilities, the break up mechanisms and the polydisperse cloud of droplets. With a Direct Numerical Simulation approach, all the scales of the system must be solved. This fine resolution is computationally very expensive and not yet possible for any flow configuration. This is the reason why reduced-order models are interesting. In this approach, the equations are solved at a larger scale (than the finest scales of the problem) and they take into account a description of the effects of the sub-scale mechanisms. For two-phase flow configurations, some reduced-order models can be used in zones of the flow where the phases are separated, while others are better adapted to disperse-phase regimes. Up to now, there does not exist any reduced-order model, that enables to describe all two-phase flow regimes.

The contributions, that have been presented in this thesis address three aspects of simulations of the atomization process. First, these contributions are part of the development of a unified modeling approach for separated phases and disperse-phase flows using reduced-order models. Second, a discretization strategy for the simulation of the two-phase flow models has also been studied and tested. Finally, an Adaptive Mesh Refinement (AMR) code has been developed and enables highly parallel computations.

Let us recall that, among the methods used to simulate separated-phase flows, mixture two-phase models may be associated with an interface capturing method, without any interface reconstruction (as in Grenier, Vila, and Villedieu (2013); Bernard-Champmartin and De Vuyst (2014) among others). The method is assumed to converge to the discontinuous locus of the interface separating both phases when the resolution is finer. The polydisperse clouds of droplets are however considered in a statistical way as a collection of objects with a spherical topology (see Kah (2010) among many others). First relations between these two approaches may appear when noting, on a one hand, that several two-fluid models are used to describe disperse bubbly flows (Drew and Passman (1999); Gavrilyuk and Saurel (2002)) and present similarities with the mixture models. These two-fluid models may also be endowed with an equation on the interface area density (Drew (1990); Morel (2015); Devassy, Habchi, and Daniel (2015)), that provides further information about the sub-scale interfacial structures. However, these models are often made of many terms issued from the averaging procedure and that involve closure issues. On the other hand, the fractional moments model of Essadki (2017) considers moments of a number density function that may have a geometrical interpretation. This encourages to consider two-phase flows configurations that are more general than spherical particles. In our approach, we propose to use Hamilton's variational principle to enrich the mixture models with only some selected sub-scale effects, useful for the description of a disperse flow of bubbles. We have shown that the models obtained with this approach are actually compatible with a description of bubbly flows, through two well-identified parameters, a micro-inertia ν and a micro-viscosity ϵ. Furthermore, these parameters control the asymptotic relations between a hierarchy of mixture models, from the most enriched model, where $\nu > 0$ and $\epsilon > 0$, to the model involved in the interface capturing strategy, where $\nu \to 0$ and $\epsilon \to 0$. Now, from the study of the geometrical description of surfaces, we can propose new paths for introducing more sub-scale information within these mixture models, based on geometrical quantities, with the

objective of describing more general interface topologies. Moreover, the relation between the statistical description of surfaces and the disperse-phase models for spherical bubbles is established in Essadki, Drui, de Chaisemartin, Larat, Ménard, and Massot (2017). One can see that all these developments participate in the convergence of both descriptions for separated phases and disperse-phase flows, with many possible improvements that are mentioned hereafter.

The discretization choices also represent a fundamental aspect of numerical simulations. These choices must be done so that the equations of the models are accurately resolved, whatever the regime of the flow, *i.e.* in the limits of small values for ν and ϵ, in the low-Mach number limit, etc. Moreover, the discretization choices should preferably be computationally cheap and compatible with parallel computations. To satisfy these two aspects, we have chosen a discretization strategy made of an operator splitting method. Then, standard finite volume schemes, adapted to an AMR strategy, are used to solve the convective part of the equations and ODE solvers are used for the integration of the source terms. The standard approaches need several modifications, due to some specificities of the two-phase mixture equations, that are: the density and sound velocity large ratios between both phases, the compressible nature of the equations and the stiffness of the source terms due to their non-linear character. Let us recall that these difficulties have been addressed in the following way:

- in cases of sharp gas-liquid interfaces, the ratio of density across the interface may be of the order of $1/1,000$ and the ratio of acoustic impedance (the product of the density by the sound velocity) of nearly $2/10,000$. In this situation, Suliciu's relaxation solver (Suliciu (1990); Bouchut (2004)), that shows the interesting properties of accuracy and robustness, requires a discretization time step that is much shorter than other approximate Riemann solvers, like HLLC (Toro (2009)) for example. This shortcoming may lead to an unnecessary high number of time iterations, which is computationally expensive. With the will of preserving the good properties of the numerical scheme, we have proposed to consider a correction of Suliciu's scheme inspired by Bouchut (2004) for the case of Euler equations close to a vacuum state. While a more rigorous adaptation of this correction to the two-phase systems is still to be investigated, the correction proposed in the Relax-B scheme shows a real improvement compared to Suliciu's scheme with respect to the computational cost,
- the discretization of the equation on the volume fraction also requires some care. Indeed, when this discretization is not able to maintain uniform pressure and velocity profiles, it may generate instabilities in the numerical solution, as discussed in Abgrall (1996). Then, we choose a standard upwind discretization for this equation and show that the unique pressure and velocity variations in the numerical tests are caused by the round-off errors when the pressures values are updated using the state laws. These errors are not amplified throughout the simulation,
- in the asymptotic limit $\epsilon \to 0$, the solutions obtained with the Relax-B scheme applied on the 4-equation and 5-equation models present a very large numerical dissipation compared to the numerical dissipation of the equilibrium 3-equation system solved with the Relax-B scheme. The characteristic sound velocities used to compute the numerical fluxes seem to be at the origin of this difficulty, since they are independent of the value of the parameter ϵ. However, the acoustic phenomena strongly depend on this parameter and are not well reproduced with the standard schemes. We propose a first modification of the scheme that considers the acoustic phase velocity obtained with the dispersion relations rather than the characteristic velocity of the convective part of the equations. This correction enables to recover the correct asymptotic behaviors of the system, but still suffers from stability issues. We think that a modification of the numerical schemes using this idea could be developed in a more rigorous way and ensure then a less dissipative asymptotic behavior of the solution,
- in the low-Mach regime, it is shown that many Godunov-type schemes suffer from a dramatically high numerical dissipation (Dellacherie (2010)) and this issue is even emphasized on adaptive grids, as is observed in sloshing (Grenier, Vila, and Villedieu (2013)) or dam-break problems (see Chapter 10). We propose two cures to this problem. The first one is based on a scheme modification inspired by Chalons, Girardin, and Kokh (2016) and applied to Relax-B scheme. The second cure is the change of quadrangular grids to triangular grids, that were shown to provide correct results in some configurations (Dellacherie (2010)). Both cures have been assessed in different low-Mach tests (a rising bubble and dam-break cases) where they show effective improvements of the numerical

results. However, both low-Mach treatments generate some numerical artifacts, either checkerboard modes or overshoots of pressure when the grid is dynamically adapted,

- a series of first tests using standard high-order and stiffly accurate methods for integrating the 5-equation source term shows that this term can be extremely stiff and lead to the failure of these solvers. Consequently, we have been interested in the design of ODE solvers dedicated to the specific source terms of the two-phase mixture models. For both the 4-equation and 5-equation models, we propose to use the exact solution of an ODE that approximates the original source term. Endowed with a clipping procedure, these approximate solvers provide solutions that stay within the admissible values for the volume fraction and that tend towards the equilibrium state in long integration times.

The choice of using cell-based AMR was initially determined by the possibility of using the p4est library (Burstedde, Wilcox, and Ghattas (2011)) that shows good performance results on massively parallel architectures. Furthermore, AMR techniques are well-suited for flows configurations with localized sharp interfaces, as in separated-phase flows for example. Using AMR implies a certain number of other choices with respect to the numerical methods. First, we propose to adapt the standard finite volume methods for Cartesian grids to the non-conforming grids, by using their regular aspect. This choice enables to use dimensional splittings and 1D slope reconstructions for a second-order MUSCL-Hancock scheme. Second, the AMR approach necessitates to choose a refinement criterion, that is compared with a refinement threshold. In our work, we have considered a refinement criterion based on the first-order variations of the solution, and more particularly on the variations of some of the variables like the volume fraction. This choice enables to increase the resolution near the interfaces while compressing the other spatial zones. The value of the refinement threshold must be adjusted according to the simulated test and to the grid refinement level. Finally, a choice must be made on the way the solution is updated when the grid is modified (refined or coarsened). In our strategy, when a cell is refined, we simply copy the value of the old cell into each of the new ones. When a set of cells are coarsened, we compute the average of the conservative variables to set the solution in the new coarser cell. These choices have been made for their simplicity and have shown good results for cases where comparisons with experimental data were possible. In particular, the whole discretization strategy presented above enables to simulate dam-break cases, which are typical low-Mach flows, using the compressible 3-equation model, discretized with Relax-B scheme on an AMR triangular grid and with a dynamic refinement of the grid near the gas-liquid interface. Although this flow configuration may appear far from the atomization configuration, it represents an interesting test of our numerical choices where they are known to possibly having difficulties. Let us finally recall that the numerical schemes, that have been presented in this thesis, have been designed for barotropic homogeneous (one velocity) models. However, all the modifications and propositions of modifications we have mentioned are probably relevant for more complex two-phase systems. In particular, they could be applied to the future enriched two-phase models.

The last aspect of numerical simulations we have been interested in and that appears to be essential for future simulations of the atomization process, is the performance of the computing code in which the previously mentioned models and schemes are integrated. The compatibility of the AMR techniques with highly parallel computations is especially a difficult task. We have chosen to use the p4est library (Burstedde, Wilcox, and Ghattas (2011)), that has the advantages of:

- providing a user-friendly framework to develop AMR applications while being application-free,
- being scalable up to several hundreds of thousands cores,
- proposing a strategy to deal with complex domain geometries.

Based on this library, we have developed the CanoP code, which is a finite volume applicative layer of p4est. CanoP has also been developed to be a framework for different applications about fluid dynamics. While still being under development, the first performance tests, among which the tests of the 3D advection of a volume fraction profile solved with Relax-B scheme, show good results. Moreover, the CanoP code has been developed so as to facilitate the integration of new applications, by using generic data structures and by proposing a set of features, related to the finite volume methods or to the AMR strategies. This will help the development and the testing of the future new reduced-order models.

Future developments

The results that have been obtained in the three main axes, that are the modeling, the numerical and the computational axes, are encouraging and justify the different choices that have been made.

The mixture two-phase models show interesting mathematical properties that make them suitable for accurate numerical simulations. By using the same methodology of derivation, namely by using Hamilton's variational principle and the second principle of thermodynamics, we can envision to further enrich these models. Three main aspects may be considered:

- according to the study of the interface geometrical properties and the associated surface density function, we could consider new energy terms that account for the mean pulsations of deformed bubbles, or for mean effects of surface tension. Therefore, these terms would involve the first order moments of the surface density function. The model would enable to describe more general flow topologies and would be closer to the kinetic models for the droplets,
- in our modeling approach, many strong assumptions have been made, like the kinematic equality between the phases and the barotropic state laws. With the objective of a full coupling with the disperse phase, where the velocity of the droplets is a priori different from the gas velocity, it will be necessary to discard these assumptions and let each phase has its own velocity and temperature (Cordesse (2019)). As in Gavrilyuk and Saurel (2002), the variational approach could be used to derive more comprehensive two-fluid models,
- within the equations for the statistical description of the interfaces, many terms still have to be closed properly.

Moreover, one will have to deal with compressibility issues. Indeed, while droplets are often assumed to be incompressible, the compression effects in the bubbles pulsations play an important role, as we have seen with the dispersion of acoustic waves.

The same numerical strategy could be considered to discretize these new models. Some features of the numerical methods should however be improved. In particular, the following points deserve further investigations:

- the Relax-B scheme must be more rigorously adapted to the two-phase systems. Indeed, it is inspired by Bouchut (2004), where it is applied to barotropic Euler equations. Therefore, the conditions that are imposed to ensure the entropic character of the scheme and to compute proper acoustic waves velocities are based on pressure laws that depend on the density only. However, in the two-phase case, the total pressure of the system also depends on the mass fractions and on the volume fractions of the phases. The conditions proposed by Bouchut (2004) should be reconsidered,
- a correction to the Relax-B scheme should also be developed to rectify the wrong numerical dissipation in the asymptotic limit $\epsilon \to 0$. Indeed, the solution we have proposed is not stable, but it could inspire a more robust approach,
- the integration strategy for the source terms could be further studied. In particular, based on the approximate solutions we propose, higher-order strategies could probably be proposed, or strategies finer than the clipping procedure could be envisioned. These improvements may probably be useful for the future two-phase models, since they may contain new relaxation terms,
- the specific low-Mach strategies also have some shortcomings and necessitate further investigation. Especially, we could envision an AMR strategy that prevents the apparition of the pressure overshoots when the grid is refined and coarsened. Then, it would be interesting to go further in the simulations of the dam-break cases and compare the compressibility effects that our method enables to describe and that are not reproduced by methods using incompressible equations.

Moreover, the accuracy of the numerical results could also be increased in several ways. First, an improvement that does not need much modification of the data structures in the CanoP code consists in discretizating the equation on the volume fraction by using the low-diffusive scheme of Després and Lagoutière (2001); Kokh and Lagoutière (2010). This way, the numerical diffusion near interfaces could be reduced. However, this method has never been tested on AMR grids and we do not know whether both aspects are compatible. Otherwise, high-order numerical schemes may be considered in the future simulations. A MOOD method (Clain, Diot, and Loubère (2011)) and a Discontinuous Galerkin scheme

(Cockburn and Shu (1989)) are currently being integrated in the CanoP code. Both approaches require some development in CanoP so as to better manage the heavy data memory allocation and enable a simpler access to large stencils of cells. The improvement of the numerical accuracy also requires to control the compression error generated by the computations on adapted grids. More appropriate refinement strategies could be considered:

- still in an AMR approach, one could use criteria based on the local truncation error as in Berger and Oliger (1984) or on smoothness indicators provided by a high-order numerical method as in Freret, Ivan, Sterck, and Groth (2017),
- one could consider using a multiresolution approach (Harten (1994); Duarte (2012)) for a mathematical control of the compression error using the multilevel analysis of the solution. However using such methods in the CanoP code will probably require some further code developments. Indeed, the multiresolution approach needs a large stencil of cells in its prediction step. Whereas it can be adapted to shared memory architectures and manycore processors, it is not yet well-suited for distributed memory architectures.

Finally, let us mention that the perspectives of development and of application for the CanoP code are numerous. First, the current developments aim at a better management of the memory allocations and memory access for the user's heavy data. This will facilitate the use of data structures that may vary in size, like in a coupling strategy between models or for the adaptivity of the Discontinuous Galerkin methods in polynomial order. Second, developments of methods for enabling the computation over heterogeneous architectures is considered. Indeed, the integration of the source terms in the kinetic-based model requires arithmetic intense computations, that may take a long time on traditional processors. To this purpose, task-driven programming environment has been tested in Essadki, Jung, Larat, Pelletier, and Perrier (2017). Finally, new applications, related to astrophysics simulations, will be integrated in the CanoP code.

Appendix A

Exact Riemann's solutions for the two-fluid models

In this appendix, we first study the mathematical properties of the three two-phase models derived in Chapter 1. We then provide the exact resolution of Riemann problems for these models. Finally, a few classic Riemann problems are introduced and the solution for three systems are computed.

A.1 General system of equations

A.1.1 Equations and mathematical properties

Let us consider the two-fluid systems under the general form in 1D

$$\left\{\begin{array}{rcrcl} \partial_t\left(\rho u\right) & + & \partial_x\left(\rho u^2+P\right) & = & 0 \\ \partial_t \rho & + & \partial_x\left(\rho u\right) & = & 0 \\ \partial_t \mathbf{V} & + & u\partial_x \mathbf{V} & = & 0 \end{array}\right. \tag{A.1}$$

where we have discarded the source terms. In its quasilinear form, and choosing the set of variables $\mathbf{U}=\left(u,P,^T\mathbf{V}\right)^T$, system (A.1) is equivalent to:

$$\left\{\begin{array}{rcrcl} u_t & + & u u_x+\frac{1}{\rho}P_x & = & 0 \\ P_t & + & uP_x+\rho c(\rho,\mathbf{V})^2 u_x & = & 0\,. \\ \mathbf{V}_t & + & u\mathbf{V}_x & = & 0 \end{array}\right. \tag{A.2}$$

We recall that the eigenvalues of system (A.2) are:

$$\lambda_1=u-c\,,\quad \lambda_2=u+c\,,\quad \lambda_3=u\,.$$

If we introduce the vector of variables $\mathbf{W}=\left(u,\rho,\mathbf{V}^T\right)^T$, the eigenvectors associated with the two first eigenvalues are respectively given by:

$$\mathbf{P}_1=\left(-c/\rho,1,^T\mathbf{0}\right)^T\,,\quad \mathbf{P}_2=\left(c/\rho,1,^T\mathbf{0}\right)^T\,.$$

Let us now suppose that $\mathbf{V}\in\mathbb{R}^N$, $N>0$, and let us write $\gamma_i=\frac{\partial P}{\partial \mathbf{V}_i}$, $i\in[\![0,N]\!]$. We can show that the eigenvectors associated with λ_3 write:

$$\mathbf{P}_{3,i}=\left(0,\frac{\delta_{i=1}}{c^2},\left(\frac{\delta_{i=j+1}}{\gamma_j}\right)^T_{j<N},-\frac{1}{\gamma_N}\right)^T\,.$$

Finally, one can easily show that P_1, P_2 and $(P_{3,i})_{i\leqslant N}$ are linearly independent.

Let us also study here the nature of the waves generated by the three eigenvalues and show that:

- the fields associated with λ_1 and λ_2 are genuinely non linear, and thus the waves are shocks or rarefactions;
- the fields associated with λ_3 are linearly degenerate and the waves are contact discontinuities.

According to Leveque (2004), the nature of the characteristic fields is given by the behavior across the phase space Ω of

$$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_j \cdot \mathbf{P}_j, \quad j \in [\![1, N+2]\!], \tag{A.3}$$

where $\boldsymbol{\nabla}_{\mathbf{W}}$ means the gradient vector obtained by differentiating by each component of vector $\mathbf{W}$. If (A.3) has a constant sign and does never cancel, then the associated field is genuinely non linear. If (A.3) is always null, then the field is linearly degenerate. For the three eigenvalues λ_1, λ_2 and λ_3, one obtains:

- for λ_1:

  $$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_1 = \left(1,\, -\partial_\rho c,\, (-\boldsymbol{\nabla}_{\mathbf{V}} c)^T\right)^T,$$

  and then one has

  $$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_1 \cdot \mathbf{P}_1 = -\frac{c}{\rho} - \frac{\partial c}{\partial \rho} < 0,$$

  according to property (4.24) ;
- for λ_2:

  $$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_2 = \left(1,\, \partial_\rho c,\, (\boldsymbol{\nabla}_{\mathbf{V}} c)^T\right)^T,$$

  and then one has

  $$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_2 \cdot \mathbf{P}_2 = \frac{c}{\rho} + \frac{\partial c}{\partial \rho} > 0,$$

  according to property (4.24) ;
- for $\lambda_{3,i}$, $i \in [\![1, N]\!]$:

  $$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_{3,i} = \left(1,\, 0,\, {}^T\mathbf{0}\right)^T,$$

  and then one has

  $$\boldsymbol{\nabla}_{\mathbf{W}}\lambda_{3,i} \cdot \mathbf{P}_{3,i} = 0.$$

The nature of the fields and waves for the Riemann problem is thus deduced from these three relations.

A.1.2 Integral curves

Let us consider the following Riemann's problem for system (A.2):

$$\mathbf{U}(0,x) = \begin{cases} \mathbf{U}_L & \text{if} \quad x < 0 \\ \mathbf{U}_R & \text{if} \quad x \geqslant 0 \end{cases} \tag{A.4}$$

The solutions to this problem are self-similar: $\mathbf{U}(t,x) = \mathbf{U}(\lambda t, \lambda x)$ for $\lambda > 0$. Let us then introduce the variable $\xi = x/t$ and $\dot{\mathbf{U}} = d\mathbf{U}/d\xi$. System (A.2) can be written as:

$$\begin{cases} -\xi \dot{u} & + & u\dot{u} + \frac{1}{\rho}\dot{P} & = & 0 \\ -\xi \dot{P} & + & u\dot{P} + \rho c(\rho, \mathbf{V})^2 \dot{u} & = & 0 \\ -\xi \dot{\mathbf{V}} & + & u\dot{\mathbf{V}} & = & 0 \end{cases}. \tag{A.5}$$

or also under the form of an ordinary differential equation:

$$\left(\mathbf{A}\left(\mathbf{U}\right) - \xi \mathbf{Id}\right)\dot{\mathbf{U}} = 0 \tag{A.6}$$

Trivial solution Whatever the value of ξ different from λ_1, λ_2 and λ_3, system (A.6) admits $\dot{\mathbf{U}} = 0$ as a solution. This means that when $(\mathbf{A}(\mathbf{U}) - \xi\mathbf{Id}) \neq 0$, then the solution is composed of constant states $\mathbf{U}(\xi) = \overline{\mathbf{U}}$.

Rarefaction waves Let us study the case when $\xi = \lambda_1$ and assume that there is no singularity in the solution. The case $\xi = \lambda_2$ can be solved similarly. System (A.5) then writes:

$$\begin{cases} c\dot{u} & + & \frac{1}{\rho}\dot{P} & = & 0 \\ c\dot{P} & + & \rho c^2 \dot{u} & = & 0 \\ c\dot{\mathbf{V}} & & & = & 0 \end{cases}.$$

First, we can see that the variables $\mathbf{V}$ are constant through the wave (as $c > 0$). We then have $\mathbf{V} = \mathbf{V}_L$.

The two other equations are equal, and the relation between u and P is:

$$\dot{u} + \frac{1}{\rho c\left(P, \mathbf{V}_L\right)}\dot{P} = 0$$

As there is no singularity, we can then integrate from ξ_L, chosen such that $\mathbf{U}(\xi_L) = \mathbf{U}_L$, to ξ, and we will then write the relation between u and P as:

$$u - u_L + \int_{P_L}^{P} \frac{dP}{\rho c(P, \mathbf{V}_L)} = 0 \tag{A.7}$$

Through the waves associated with λ_2, the relation is given by:

$$u - u_R - \int_{P_R}^{P} \frac{dP}{\rho c(P, \mathbf{V}_R)} = 0 \tag{A.8}$$

Shock waves Through shock waves, there is a jump in the solution. Let us then use the Rankine-Hugoniot relations to get the solution through a shock associated with λ_1: let σ be the shock speed, and

$$\begin{aligned} \sigma[\rho u] &= [\rho u^2 + P] \\ \sigma[\rho] &= [\rho u] \\ \sigma[\rho \mathbf{V}] &= [\rho \mathbf{V} u] \end{aligned}$$

where the notation $[\cdot]$ is used to denote the jump in the variables.

Especially, through λ_1, we will get:

$$\sigma(\rho_L u_L - \rho u) = \rho_L u_L^2 + P_L - \rho u^2 - P \tag{A.9}$$

$$\sigma(\rho_L - \rho) = \rho_L u_L - \rho u \tag{A.10}$$

$$\sigma(\rho_L \mathbf{V}_L - \rho \mathbf{V}) = \rho_L \mathbf{V}_L u_L - \rho \mathbf{V} u \tag{A.11}$$

Equations (A.10) and (A.11) can be written under the form:

$$\begin{aligned} \rho_L(\sigma - u_L) &= \rho(\sigma - u) \\ \rho_L \mathbf{V}_L(\sigma - u_L) &= \rho \mathbf{V}(\sigma - u) \end{aligned}$$

and injecting the first relation in the second one, it is obvious, provided that $u \neq \sigma$, that $\mathbf{V} = \mathbf{V}_L$.

Now, from (A.10), one has the relation for the shock velocity:

$$\sigma = \frac{\rho_L u_L - \rho u}{\rho_L - \rho}$$

that, injected in (A.9), gives:

$$\begin{aligned}
(\rho_L u_L - \rho u)^2 &= (\rho_L - \rho)\left(\rho_L u_L^2 - \rho u^2\right) + (\rho_L - \rho)(P_L - P) \\
\cancel{(\rho_L u_L)^2} + \cancel{(\rho u)^2} - 2\rho\rho_L u u_L &= \cancel{(\rho_L u_L)^2} + \cancel{(\rho u)^2} - \rho\rho_L\left(u_L^2 + u^2\right) + (\rho_L - \rho)(P_L - P) \\
\rho\rho_L\left(u - u_L\right)^2 &= (\rho_L - \rho)(P_L - P)
\end{aligned}$$

This gives:

$$u = u_L \pm \sqrt{\frac{(\rho_L - \rho)(P_L - P)}{\rho\rho_L}}$$

And finally, for a left shock where the resulting pressure P must be larger than the left pressure P_L:

$$u = u_L - \sqrt{\frac{(\rho_L - \rho)(P_L - P)}{\rho\rho_L}} \tag{A.12}$$

Similarly, we will have through the 2-wave $\mathbf{V} = \mathbf{V}_R$ and:

$$u = u_R + \sqrt{\frac{(\rho_R - \rho)(P_R - P)}{\rho\rho_R}} \tag{A.13}$$

Contact discontinuity By definition of the contact discontinuity, $\lambda_3 = u$ is a Riemann invariant for this wave. Then, by using the jump relations, one has $\sigma = u$ and $[P] = 0$: P is also a Riemann's invariant here.

A.1.3 Solution to the Riemann's problem

Let us first summarize the curves that relate u and P through the 1- and 2-waves.

- For $\lambda_1 = u - c(P, \mathbf{V})$:

$$u_1(\xi) = u_L - \begin{cases} \displaystyle\int_{P_L}^{P(\xi)} \frac{dP}{\rho c(P, \mathbf{V}_L)} & \text{if} \quad P(\xi) \leqslant P_L \\ \displaystyle\sqrt{\frac{(\rho_L - \rho(\xi))(P_L - P(\xi))}{\rho(\xi)\rho_L}} & \text{if} \quad P(\xi) > P_L \end{cases} \tag{A.14}$$

- For $\lambda_2 = u + c(P, \mathbf{V})$:

$$u_2(\xi) = u_R + \begin{cases} \displaystyle\int_{P_R}^{P(\xi)} \frac{dP}{\rho c(P, \mathbf{V}_R)} & \text{if} \quad P(\xi) \leqslant P_R \\ \displaystyle\sqrt{\frac{(\rho_R - \rho(\xi))(P_R - P(\xi))}{\rho(\xi)\rho_R}} & \text{if} \quad P(\xi) > P_R \end{cases} \tag{A.15}$$

The Riemann's solution is made of 4 constant states separated by the three waves related to λ_1, λ_2 and λ_3. According to the Riemann's invariant through the contact discontinuity, one knows that the intermediate states are given by:

$$\mathbf{U}_L^* = \begin{pmatrix} u^* \\ P^* \\ \mathbf{V}_L \end{pmatrix} \quad \text{and} \quad \mathbf{U}_R^* = \begin{pmatrix} u^* \\ P^* \\ \mathbf{V}_R \end{pmatrix}$$

where P^* can be found using the equality (A.14) = (A.15), then u^* is estimated using (A.14) or (A.15) where $P(\xi) = P^*$.

Remark 39. *In all these relations, we suppose that we have $\rho = \rho(P, \mathbf{V})$, with some good properties.*

The Riemann's solution is then given by:

- if the left wave is a rarefaction wave:
  - if $\xi < u_L - c_L$:
$$\mathbf{U}(\xi) = \mathbf{U}_L$$
  - if $u_L - c_L \leqslant \xi < u^* - c(P^*, \mathbf{V}_L)$:
$$\mathbf{U}(\xi) = \begin{pmatrix} u_1(\xi) \\ p(\xi) \\ \mathbf{V}_L \end{pmatrix} \quad \text{s. t. } \xi = u(\xi) - c(P(\xi), \mathbf{V}_L)$$
  - if $u^* - c(P^*, \mathbf{V}_L) \leqslant \xi < u^*$:
$$\mathbf{U}(\xi) = \mathbf{U}_L^*$$
- if the left wave is a shock wave, then $\sigma_L = \frac{\rho_L u_L - \rho_L^* u^*}{\rho_L - \rho_L^*}$ and:
  - if $\xi < u_L - \sigma_L$:
$$\mathbf{U}(\xi) = \mathbf{U}_L$$
  - if $u_L - \sigma_L \leqslant \xi < u^*$:
$$\mathbf{U}(\xi) = \mathbf{U}_L^*$$
- if the right wave is a rarefaction wave:
  - if $u^* \leqslant \xi < u^* + c(P^*, \mathbf{V}_R)$:
$$\mathbf{U}(\xi) = \mathbf{U}_L^*$$
  - if $u^* + c(P^*, \mathbf{V}_R) \leqslant \xi < u_R - c_R$:
$$\mathbf{U}(\xi) = \begin{pmatrix} u_3(\xi) \\ p(\xi) \\ \mathbf{V}_R \end{pmatrix} \quad \text{s. t. } \xi = u(\xi) + c(P(\xi), \mathbf{V}_R)$$
  - if $u_R + c_R \leqslant \xi$:
$$\mathbf{U}(\xi) = \mathbf{U}_R$$
- if the right wave is a shock wave, then $\sigma_R = \frac{\rho_R u_R - \rho_R^* u^*}{\rho_R - \rho_R^*}$ and:
  - if $u^* \leqslant \xi < u_R + \sigma_R$:
$$\mathbf{U}(\xi) = \mathbf{U}_R^*$$
  - if $u_R + \sigma_R \leqslant \xi$:
$$\mathbf{U}(\xi) = \mathbf{U}_R$$

A.2 Application to the two-fluid systems

A.2.1 3-equation system

The 3-equation system can be written under the form:

$$\left\{\begin{array}{lcccl} u_t & + & uu_x + \dfrac{1}{\rho(p,Y)}p_x & = & 0 \\ p_t & + & up_x + \rho(p,Y)c(p,Y)^2u_x & = & 0 \\ Y_t & + & uY_x & = & 0 \end{array}\right. . \tag{A.16}$$

with:

- the pressure given by the pressure equality between the two fluids $p = p_1 = p_2$,
- the characteristic sound velocity $c(p,Y)$ which is Wood's sound velocity:

$$c(p,Y) = \left(\rho(p,Y)\left(\frac{\alpha(p,Y)}{\rho_1(p)c_1^2} + \frac{1-\alpha(p,Y)}{\rho_2(p)c_2^2}\right)\right)^{-1/2}$$

- the vector $\mathbf{V}$ is $\mathbf{V} = Y$.

The functions $\rho_1(p)$ and $\rho_2(p)$ are give by the fluids state laws, and $\alpha(p,Y)$ by:

$$\alpha(p,Y) = \frac{Y\rho_2(p)}{Y\rho_2(p) + (1-Y)\rho_1(p)}$$

Finally, $\rho(p,Y)$ writes:

$$\rho(Y,p) = \alpha(Y,p)\rho_1(p) + (1-\alpha(Y,p))\rho_2(p).$$

We consider here barotropic state laws (for $k = 1, 2$):

$$\rho_k(p) = \rho_k^o + \frac{1}{c_k^2}\left(p - p^o\right).$$

For the 3-equation system, formulae (A.14) and (A.15) can not be written in an explicit form, as there is no prime function to the function

$$(p,Y) \mapsto \frac{1}{\rho(p,Y)c(p,Y)}.$$

In practice, one will have to use approximations for the formulae (A.14) and (A.15), and root-finding algorithms to compute p^*...

A.2.2 4-equation system

For this system, the exact solution to the Riemann's problem can also be found in Chanteperdrix, Villedieu, and Vila (2002).

Let us recall here the properties of this system of equations. First, it can be written under the form (without the source term):

$$\left\{\begin{array}{lcccl} u_t & + & uu_x + \dfrac{1}{\rho(p,Y,\alpha)}p_x & = & 0 \\ p_t & + & up_x + \rho(p,Y,\alpha)c(p,Y,\alpha)^2u_x & = & 0 \\ Y_t & + & uY_x & = & 0 \\ \alpha_t & + & u\alpha_x & = & 0 \end{array}\right. . \tag{A.17}$$

Remark 40. *This system deprived from its source terms has no physical meaning. Taking the source terms into account would modify the differential equation (A.6) and exact solutions would then be much more complex to find.*

With the 4-equation system, α is now an independent variable. The system gets the following properties:

- there exists two fluid pressures p_1 and p_2 that are no longer equal. They are related to p through the relation:

$$p = \alpha p_1 + (1-\alpha)p_2.$$

- The characteristic sound velocity is given by the Frozen sound velocity formula:

$$c(p, Y, \alpha) = \sqrt{Y c_1^2 + (1-Y)c_2^2}$$

- the vector $\mathbf{V}$ is now composed of $\mathbf{V} = (Y, \alpha)^T$.

Let us assume that the pressure laws are linearized barotropic pressure laws, given by:

$$\rho_k : p_k \mapsto \rho_k^o + \frac{1}{c_k^2}\left(p_k - p^o\right).$$

For the set of variables chosen to write system (A.17), the density is given by:

$$\rho(p, Y, \alpha) = \alpha\rho_1(p_1) + (1-\alpha)\rho_2(p_2)$$

Let us however change the variables, to integrate more easily the term $\frac{1}{\rho \mathrm{c}_{\mathrm{Frozen}}}$, and consider the set $(\rho, Y, \alpha)^T$. We have then:

$$\begin{aligned}\int_{p_i}^{p_e} \frac{dp}{\rho \mathrm{c}_{\mathrm{Frozen}}} &= \int_{\rho(p_i)}^{\rho(p_e)} \frac{\mathrm{c}_{\mathrm{Frozen}}^2 d\rho}{\rho \mathrm{c}_{\mathrm{Frozen}}} \\ &= \mathrm{c}_{\mathrm{Frozen}} \log\left(\frac{\rho(p_e)}{\rho(p_i)}\right)\end{aligned}$$

with the following relation between ρ and p:

$$p = \rho \mathrm{c}_{\mathrm{Frozen}}^2 + \alpha\left(p_1^o - c_1^2\rho_1^o\right) + (1-\alpha)\left(p_2^o - c_2^2\rho_2^o\right)$$

Functions (A.14) and (A.15) finally write for the 4-equation system:

- For $\lambda_1 = u - c(P, \mathbf{V})$:

$$u_1(\xi) = u_L - \begin{cases} \mathrm{c}_{\mathrm{Frozen}} \log\left(\dfrac{A^o + P(\xi)}{A^o + P_L}\right) & \text{if} \quad P(\xi) \leqslant P_L \\ \sqrt{\dfrac{(\rho_L - \rho(\xi))(P_L - P(\xi))}{\rho(\xi)\rho_L}} & \text{if} \quad P(\xi) > P_L \end{cases} \tag{A.18}$$

- For $\lambda_2 = u + c(P, \mathbf{V})$:

$$u_2(\xi) = u_R + \begin{cases} \mathrm{c}_{\mathrm{Frozen}} \log\left(\dfrac{A^o + P(\xi)}{A^o + P_R}\right) & \text{if} \quad P(\xi) \leqslant P_R \\ \sqrt{\dfrac{(\rho_R - \rho(\xi))(P_R - P(\xi))}{\rho(\xi)\rho_R}} & \text{if} \quad P(\xi) > P_R \end{cases} \tag{A.19}$$

where

$$A^o = -\alpha\left(p_1^o - c_1^2\rho_1^o\right) - (1-\alpha)\left(p_2^o - c_2^2\rho_2^o\right),$$

and the pressure laws are chosen in such a way for A^o to be positive whatever the value of α.

A.2.3 5-equation system

In the 5-equation system, two variables are transported, that represent a subscale physics. The subscale effects interact with the macro-variables in the momentum flux, through pressure. This system without the source terms writes:

$$\left\{\begin{array}{lcccl} u_t & + & u u_x + \dfrac{1}{\rho(p,Y,\alpha,w)} p_x & = & 0 \\ p_t & + & u p_x + \rho(p,Y,\alpha,w) c(p,Y,\alpha,w)^2 u_x & = & 0 \\ Y_t & + & u Y_x & = & 0 \\ \alpha_t & + & u \alpha_x & = & 0 \\ w_t & + & u w_x & = & 0 \end{array}\right. . \tag{A.20}$$

with the following properties:

- pressure p is related to ρ and to the other variables through the phasic pressures p_1 and p_2 by:

$$p = \alpha p_1 + (1-\alpha) p_2 + \frac{1}{2}(\rho Y w)^2 .$$

- the characteristic sound speed is given by the pulsation sound velocity, defined by:

$$\mathrm{c}_{\mathrm{Puls}}(p,Y,\alpha,w)^2 = \mathrm{c}^2_{\mathrm{Frozen}} + \rho(Yw)^2.$$

- The vector $\mathbf{V}$ is now composed by $\mathbf{V} = (Y,\alpha,w)^T$.

As for the 4-equation system, we first perform a change in the variables, and consider $(\rho, Y, \alpha, w)^T$.

Using the same notations as for the 4-equation system, we have the following relation between ρ and p:

$$p = -A^o + \rho\left(\mathrm{c}^2_{\mathrm{Frozen}} + \frac{1}{2}\rho(Yw)^2\right) \tag{A.21}$$

Then, the integral function becomes:

$$\begin{aligned} \int_{p_i}^{p_e} \frac{dp}{\rho \mathrm{c}_{\mathrm{Puls}}} &= \int_{\rho(p_i)}^{\rho(p_e)} \frac{\mathrm{c}^2_{\mathrm{Puls}} d\rho}{\rho \mathrm{c}_{\mathrm{Puls}}} \\ &= \int_{\rho(p_i)}^{\rho(p_e)} \frac{\sqrt{\mathrm{c}^2_{\mathrm{Frozen}} + \rho(Yw)^2}\, d\rho}{\rho} \\ &= \left[\mathrm{c}_{\mathrm{Frozen}} \log\left(\frac{\sqrt{\mathrm{c}^2_{\mathrm{Frozen}} + \rho(Yw)^2} - \mathrm{c}_{\mathrm{Frozen}}}{\sqrt{\mathrm{c}^2_{\mathrm{Frozen}} + \rho(Yw)^2} + \mathrm{c}_{\mathrm{Frozen}}} \right) + 2\sqrt{\mathrm{c}^2_{\mathrm{Frozen}} + \rho(Yw)^2} \right]_{\rho(p_i)}^{\rho(p_e)} \end{aligned}$$

Now to recover a function on p, one has to reverse relation (A.21). As it is a second order polynomial function of ρ, we can easily compute its roots, which are:

$$\rho_+ = \frac{-\mathrm{c}^2_{\mathrm{Frozen}} + \sqrt{\Delta}}{(Yw)^2}$$
$$\rho_- = \frac{-\mathrm{c}^2_{\mathrm{Frozen}} - \sqrt{\Delta}}{(Yw)^2}$$

with:

$$\Delta = \mathrm{c}^4_{\mathrm{Frozen}} + 2(A^o + p)(Yw)^2.$$

We can note that $A^o + p$ is positive, ensuring then the positivity of Δ. It is also obvious that $\rho_+ \geqslant 0$ and $\rho_- < 0$. The root ρ_+ then gives the expression of ρ as a function of p.

A.3 Exact solution to some test cases

Multiple Riemann's problems are available, and used to test numerical schemes. One can find many of them in Toro (2009) for instance. We present here some of them with the exact solution associated with the three systems of equations.

The root-finding algorithm that was used, is a simple dichotomy-based one. It is used to find the pressure *star state* by balancing the integral curves for shocks and rarefactions, depending on the initial state. It is also used then to compute the pressure value within rarefaction fans.

For the 3-equation system, where an integral function must be approximated, we have used the Simpson's rule to get a more accurate result.

A.3.1 Advection test case

The first test does not really involve the resolution of the shock and rarefaction waves. Indeed, from one side of the Riemann's problem, the variables associated with the acoustic phenomena are constant. The transported variables alone present a discontinuity. The solution is given by the displacement of this discontinuity at the fluid's velocity u_0.

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	p_0	p_0
mass fraction	Y_L	Y_R
velocity	u_0	u_0
volume fraction	α_L	α_R
vibration	w_0	w_0

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$1.0 \cdot 10^5$	$1.0 \cdot 10^5$
mass fraction	$1.0 - 1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-11}$
velocity	1.0	1.0
volume fraction	$1.0 - 1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-8}$
vibration	0.0	0.0

A.3.2 Monophasic Sod test case

Here, we choose as sound velocities $c_1 = 3.0 \cdot 10^2$ and $c_2 = 1.5 \cdot 10^3$.

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	p_L	$p_R < p_L$
mass fraction	Y_0	Y_0
velocity	u_0	u_0
volume fraction	α_0	α_0
vibration	w_0	w_0

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$5.0 \cdot 10^5$	$1.0 \cdot 10^5$
mass fraction	$4.4 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
velocity	0.0	0.0
volume fraction	0.5	0.5
vibration	0.0	0.0

A.3.3 Two-fluid shock test case

Here, we choose as sound velocities $c_1 = 3.0$ and $c_2 = 15.0$.

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	p_L	$p_R < p_L$
mass fraction	Y_L	Y_R
velocity	u_0	u_0
volume fraction	α_L	α_R
vibration	w_0	w_0

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$1.1 \cdot 10^5$	$1.0 \cdot 10^5$
mass fraction	$1.0 - 1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-11}$
velocity	0.0	0.0
volume fraction	$1.0 - 1.0 \cdot 10^{-8}$	$1.0 \cdot 10^{-8}$
vibration	0.0	0.0

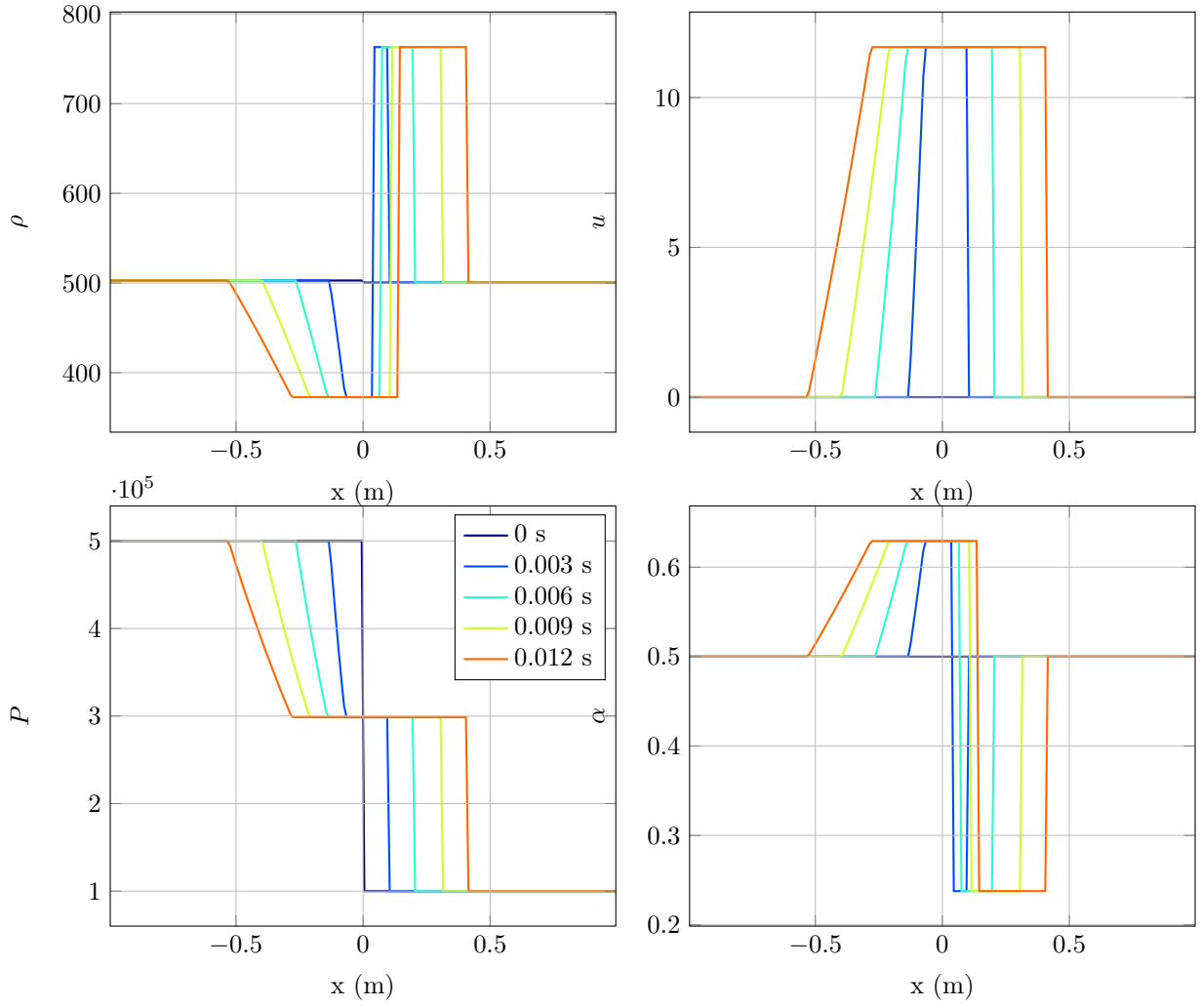


Figure A.1: *Monophasic shock solution for the 3-equation system, at different times.*

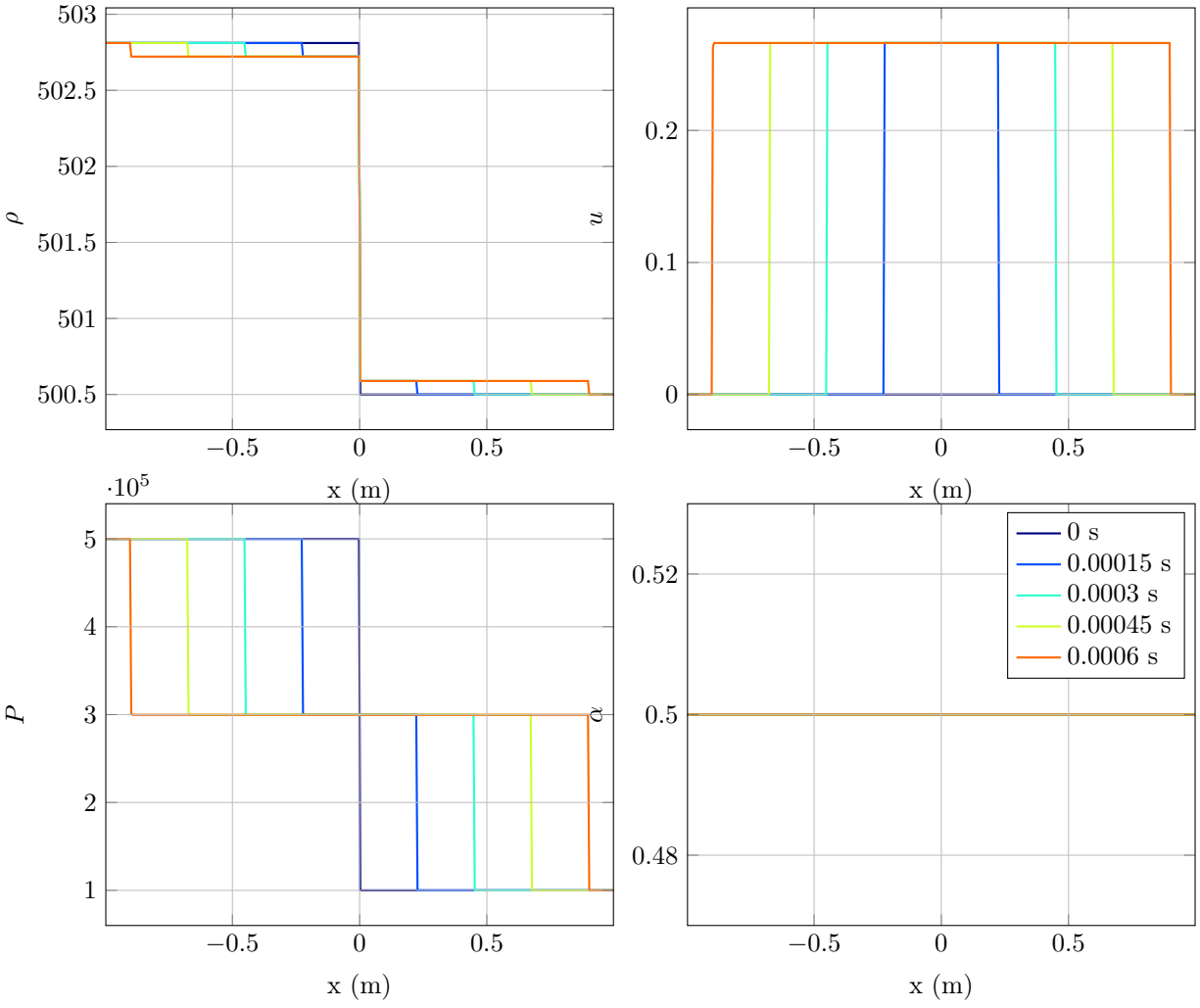


Figure A.2: *Monophasic shock solution for the 4-equation system, at different times.*

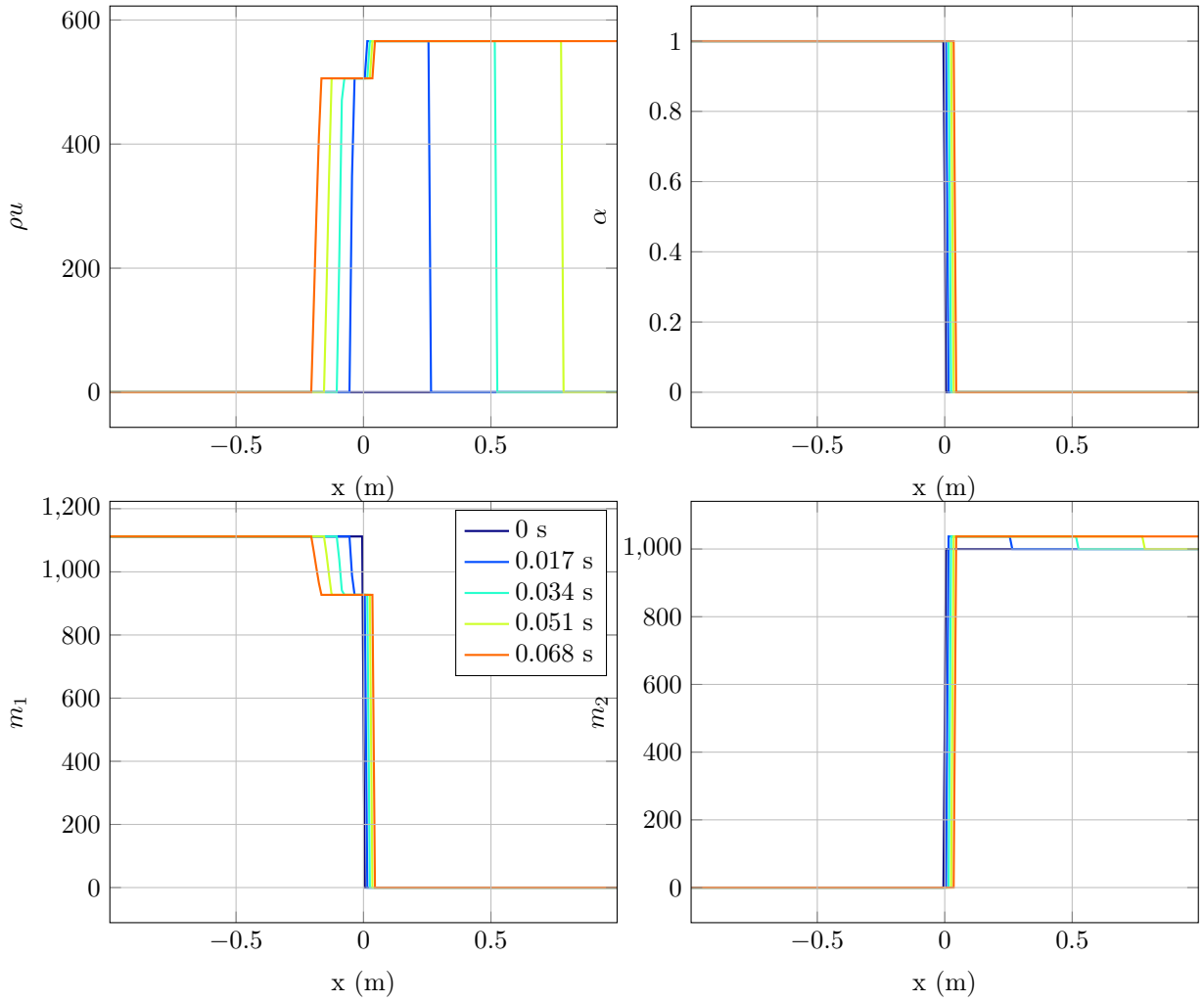


Figure A.3: *Two-fluid shock solution for the 3-equation system, at different times.*

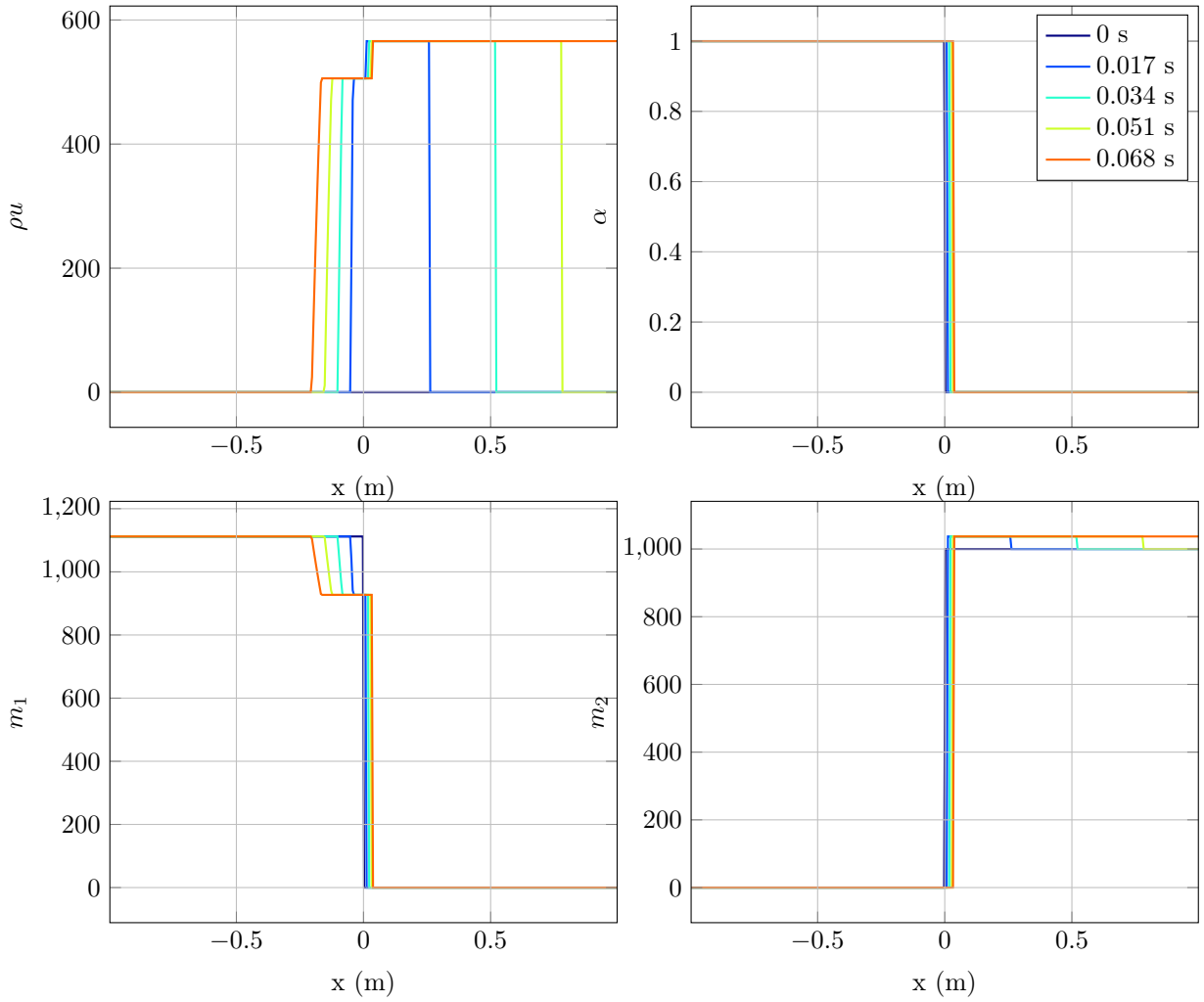


Figure A.4: *Two-fluid shock solution for the 4-equation system, at different times.*

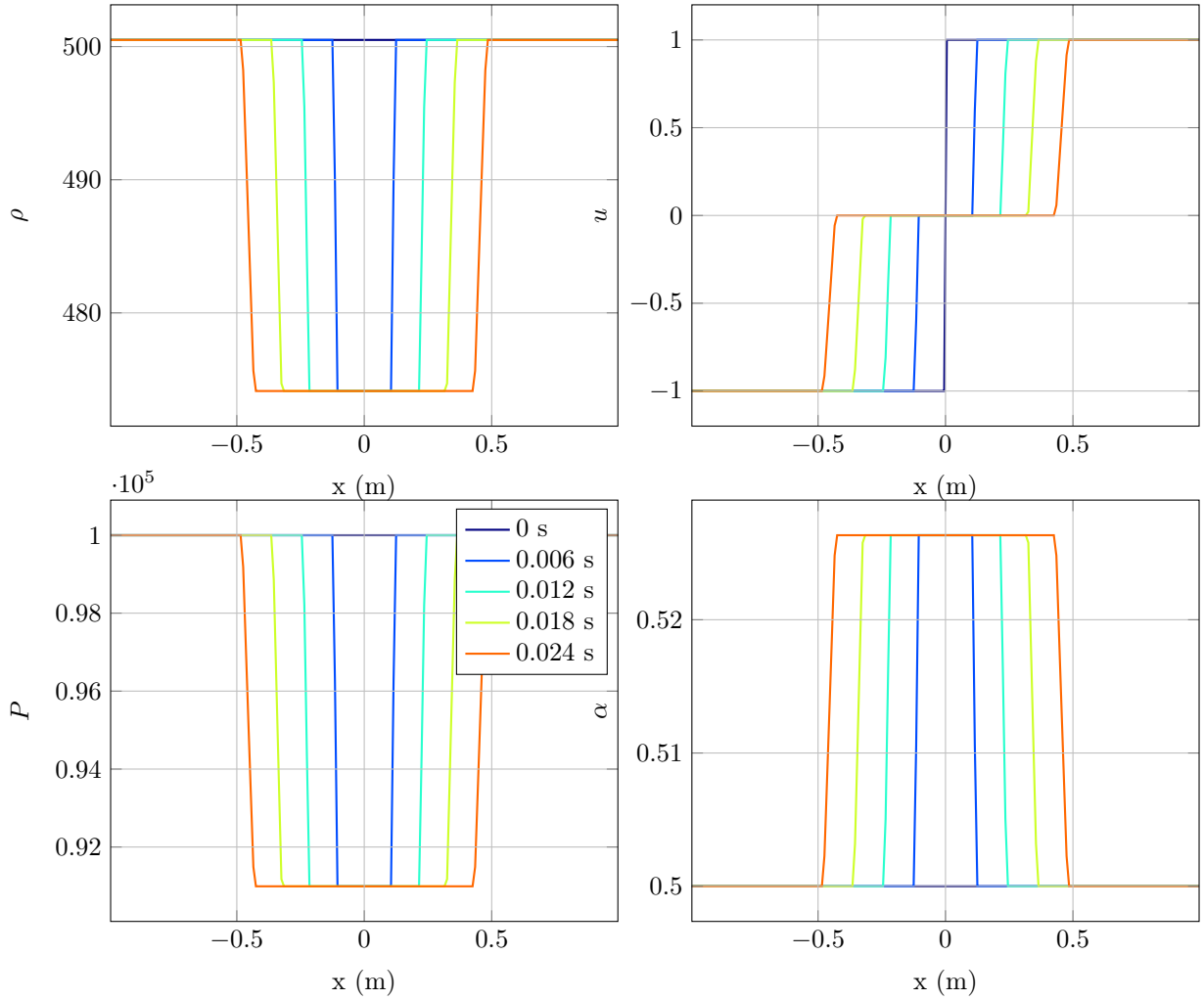


Figure A.5: *Double rarefaction solution for the 3-equation system, at different times.*

A.3.4 Double rarefaction test case

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	p_0	p_0
mass fraction	Y_0	Y_0
velocity	u_L	$u_R > u_L$
volume fraction	α_L	α_R
vibration	w_0	w_0

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$1.0 \cdot 10^5$	$1.0 \cdot 10^5$
mass fraction	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
velocity	-1.0	1.0
volume fraction	0.5	0.5
vibration	0.0	0.0

A.3.5 Pressure disequilibrium and vibrations

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	p_0	p_0
mass fraction	Y_0	Y_0
velocity	u_0	u_0
volume fraction	α_0	α_0
vibration	w_L	w_R

	W_L	W_R
location	$x < 0$	$x > 0$
pressure	$1.0 \cdot 10^5$	$1.0 \cdot 10^5$
mass fraction	$1.0 \cdot 10^{-3}$	$1.0 \cdot 10^{-3}$
velocity	0.0	0.0
volume fraction	0.5	0.5
vibration	$1.0 \cdot 10^4$	0.0

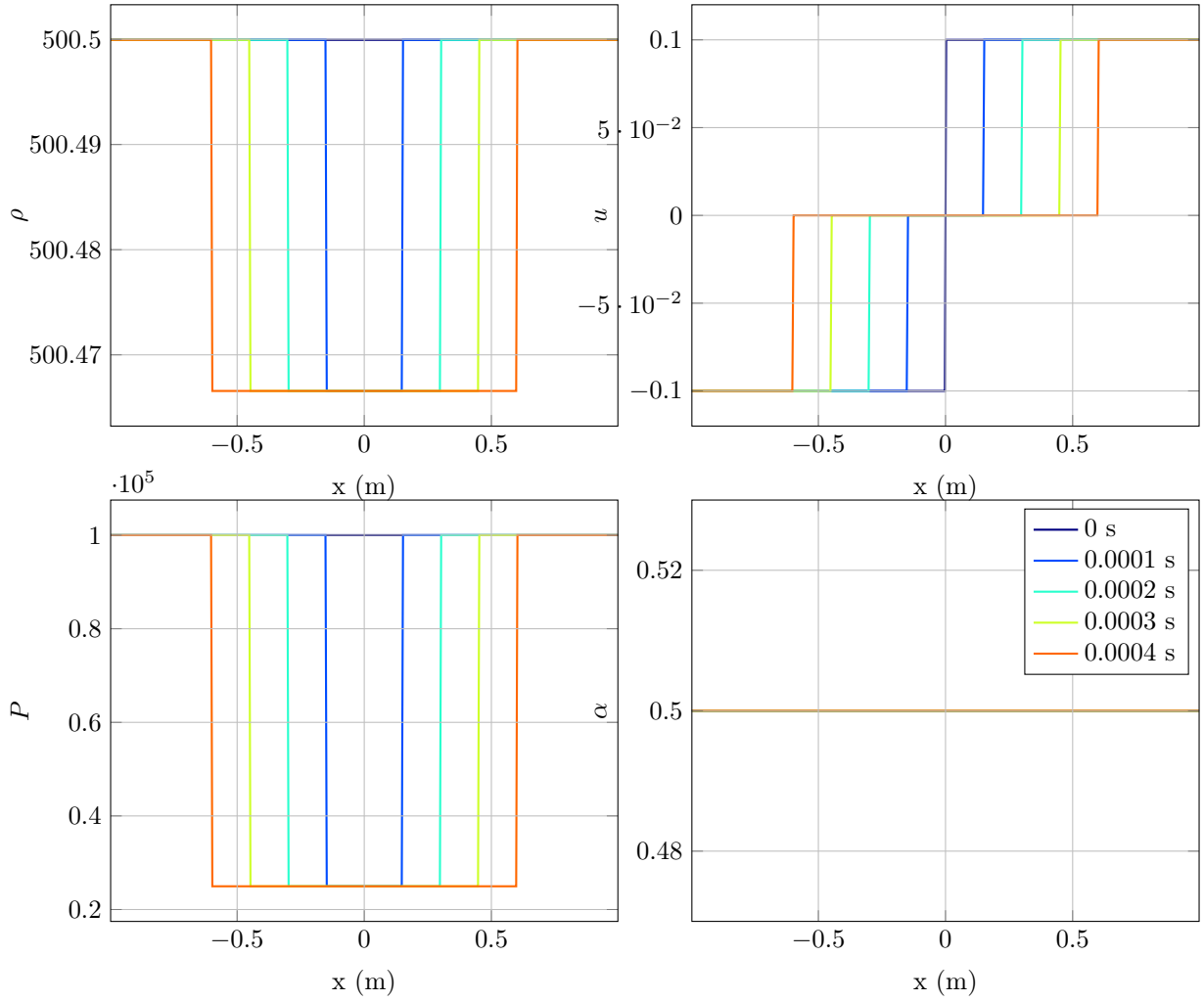


Figure A.6: *Double rarefaction solution for the 4-equation system, at different times.*

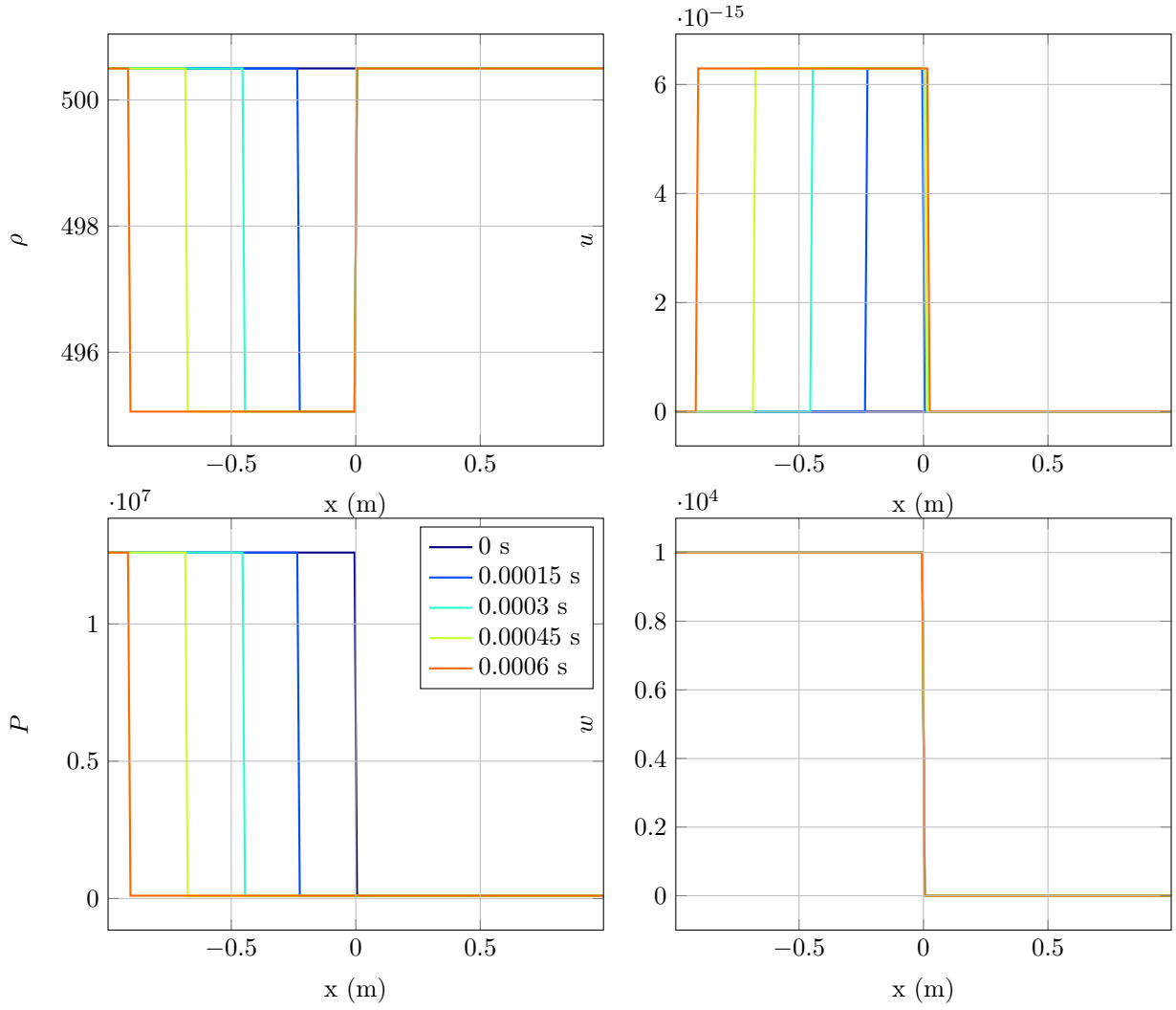


Figure A.7: *Vibrations case for the 5-equation system.*

Appendix B

On the stiffness of the source term in the 5-equation system

We present in this section tests of the integration of the source term for the 5-equation two-phase system with the Radau5 solver (Hairer and Wanner (1996); Hairer (2017)) provided by Assimulo Python package (Andersson, Führer, Akesson, and Gäfvert (2017)).

We recall that the ODE to be solved reads:

$$\begin{cases} \mathrm{d}_t\alpha &= \dfrac{\overline{\rho}\overline{Y}w}{\sqrt{\nu}} \\ \mathrm{d}_t w &= -\dfrac{\epsilon}{\nu}w \quad + \quad \dfrac{p_1(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x})) - p_2(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x}))}{\overline{\rho}\overline{Y}\sqrt{\nu}} \end{cases}$$

with:

$$\Delta p(\alpha) = p_1(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x})) - p_2(\overline{\rho},\overline{Y},\alpha(t,\mathbf{x})) = p^o + c_1^2\frac{\overline{\rho}\,\overline{Y}}{\alpha} - c_2^2\frac{\overline{\rho}\,(1-\overline{Y})}{1-\alpha}$$

where $\overline{\rho}$, $\overline{Y}$, ν, ϵ, p^o, c_1 and c_2 are constant, whose values are:

$$\overline{\rho} = 500.5, \quad \overline{Y} = 1.0\cdot 10^{-3}, \quad \nu = 1.0\cdot 10^{-4}, \quad \epsilon = 1.0,$$
$$p^o = -2.5\cdot 10^5, \quad c_1 = 3.4, \quad c_2 = 15.$$

The initial solution is given by:

$$\alpha = \alpha_0, \quad \text{and} \quad w = 0.0$$

The Jacobian matrix associated with the ODE is easily computed and reads:

$$J = \begin{pmatrix} 0 & \dfrac{\overline{\rho}\overline{Y}}{\sqrt{\nu}} \\ \Delta p'(\alpha) & -\dfrac{\epsilon}{\nu} \end{pmatrix}$$

with:

$$\Delta p'(\alpha) = -c_1^2\frac{\overline{\rho}\,\overline{Y}}{\alpha^2} - c_2^2\frac{\overline{\rho}\,(1-\overline{Y})}{(1-\alpha)^2}.$$

The Radau5 solver is then run up to time $t_f = 1.5\cdot 10^{-4}$ with the following parameters:

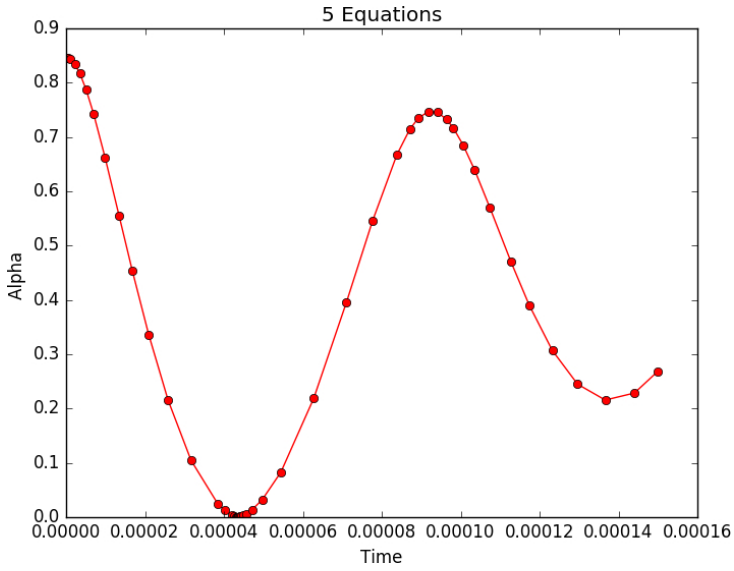
Parameter	Value	Comment
atol	$1.0\cdot 10^{-4}$	Absolute tolerance
rtol	$1.0\cdot 10^{-4}$	Relative tolerance
inith	$1.0\cdot 10^{-4}$	Initial step size
usejac	True	User-defined Jacobian

In the table below, we show the minimum time step which has been used by the solver to satisfy the required tolerance, and the number of iterations performed up to the final time t_f, for different initial values of α_0.

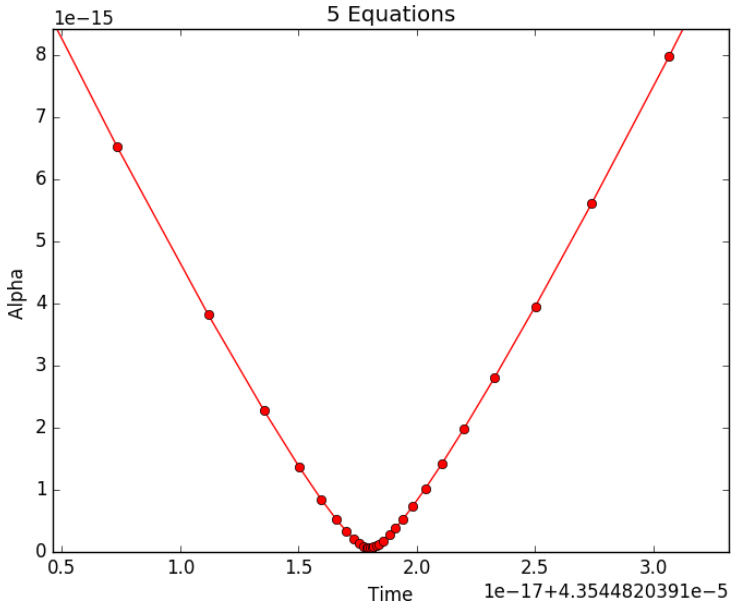
α_0	$\Delta t_{\min}$	N_{it}
0.8455	2.16261596725e-07	49
0.8456	9.35420622849e-08	52
0.8457	4.16165547752e-08	61
0.8458	2.0556396378e-10	84
0.8459	9.99737517435e-13	109
0.846	4.26982532447e-15	130
0.8461	1.86889349482e-17	150
0.8462	6.09863722023e-20	168

For $\alpha_0 = 0.8463$, the computation stops with the error message: `Exception: Radau5 failed with flag -3`.

Finally, for $\alpha_0 = 0.8462$, we show in Figure B.1 the solution computed by Radau5, at the time steps represented with the circle markers.



(a) Whole solution.



(b) Zoom in the stiff zone.

Figure B.1: *Representation of the solution for α computed by Radau5, with $\alpha_0 = 0.8462$.*

Appendix C

Analysis of the truncation error in the numerical resolution of the 4-equation system

The present study is related to the study of Chapter 5, section 5.3.1. We recall that we have noticed a large difference between the numerical dissipation of the solution for the 3-equation model and the solution for the 4-equation system in the equilibrium asymptotic limit. Here, we propose a first analysis on the origins of this large dissipation.

To get an idea about the effect of the wave velocity on the numerical dispersion relations, and in particular in the numerical dissipation, let us consider here a simpler finite volume scheme with the Lax-Friedrich's flux, applied to the 4-equation system. Let us consider first a conservative variable: $W \in \{\rho, \rho Y, \rho u\}$, and $F(W)$ its associated flux. The update formula writes:

$$\begin{aligned} W_i^{n+1} &= W_i^n + \frac{\Delta t}{2\Delta x}\left[F(W_i^n) + F(W_{i-1}^n) - a_{i-1/2}\left(W_i^n - W_{i-1}^n\right)\right] \\ &\quad - \frac{\Delta t}{2\Delta x}\left[F(W_{i+1}^n) + F(W_i^n) - a_{i+1/2}\left(W_{i+1}^n - W_i^n\right)\right] \\ &= W_i^n + \frac{\Delta t}{2\Delta x}\left[F(W_{i-1}^n) - F(W_{i+1}^n)\right] - \frac{\Delta t}{2\Delta x}\left[a_{i-1/2}\left(W_i^n - W_{i-1}^n\right) - a_{i+1/2}\left(W_{i+1}^n - W_i^n\right)\right] \end{aligned}$$

This corresponds to a centered discretization of the spatial operator, stabilized by an adequate upwind diffusion term. Indeed, we can see that using Taylor expansion of the continuous variables, in the following way:

$$W_{i+1} \approx \mathcal{W}(x_{i+1}) \approx W_i + \Delta x \frac{\partial \mathcal{W}}{\partial x}(x_i) + \frac{\Delta x^2}{2}\frac{\partial^2 \mathcal{W}}{\partial x^2}(x_i) + O(\Delta x^3),$$

and in the same way, we also have:

$$\begin{aligned} W_{i-1} &\approx W_i - \Delta x \frac{\partial \mathcal{W}}{\partial x}(x_i) + \frac{\Delta x^2}{2}\frac{\partial^2 \mathcal{W}}{\partial x^2}(x_i) + O(\Delta x^3) \\ F(W_{i+1}) &\approx F(W_i) + \Delta x \frac{\partial \mathcal{F}(\mathcal{W})}{\partial x}(x_i) + \frac{\Delta x^2}{2}\frac{\partial^2 \mathcal{F}(\mathcal{W})}{\partial x^2}(x_i) + O(\Delta x^3) \\ F(W_{i-1}) &\approx F(W_i) - \Delta x \frac{\partial \mathcal{F}(\mathcal{W})}{\partial x}(x_i) + \frac{\Delta x^2}{2}\frac{\partial^2 \mathcal{F}(\mathcal{W})}{\partial x^2}(x_i) + O(\Delta x^3) \end{aligned}$$

and

$$a_{i+1/2} \approx a_i + \frac{\Delta x}{2}\frac{\partial a}{\partial x}(x_i) + \frac{\Delta x^2}{4}\frac{\partial^2 a}{\partial x^2}(x_i) + O(\Delta x^3)$$
$$a_{i-1/2} \approx a_i - \frac{\Delta x}{2}\frac{\partial a}{\partial x}(x_i) + \frac{\Delta x^2}{4}\frac{\partial^2 a}{\partial x^2}(x_i) + O(\Delta x^3)$$

We thus can approximate the different parts of the update formula with:

$$\begin{aligned}
F(W_{i+1}) - F(W_{i-1}) &\approx 2\Delta x \frac{\partial \mathcal{F}(\mathcal{W})}{\partial x}(x_i) + O(\Delta x^3) \\
a_{i-1/2}\left(W_i^n - W_{i-1}^n\right) &= a_i \frac{\partial \mathcal{W}}{\partial x}(x_i)\Delta x - \left(\frac{1}{2}\frac{\partial a}{\partial x}(x_i)\frac{\partial \mathcal{W}}{\partial x}(x_i) + \frac{1}{2}a_i\frac{\partial^2 \mathcal{W}}{\partial x^2}(x_i)\right)\Delta x^2 + O(\Delta x^3) \\
&= a_i \frac{\partial \mathcal{W}}{\partial x}(x_i)\Delta x - \frac{\partial}{\partial x}\left(a\frac{\partial \mathcal{W}}{\partial x}\right)(x_i)\frac{\Delta x^2}{2} + O(\Delta x^3) \\
a_{i+1/2}\left(W_{i+1}^n - W_i^n\right) &= a_i \frac{\partial \mathcal{W}}{\partial x}(x_i)\Delta x + \left(\frac{1}{2}\frac{\partial a}{\partial x}(x_i)\frac{\partial \mathcal{W}}{\partial x}(x_i) + \frac{1}{2}a_i\frac{\partial^2 \mathcal{W}}{\partial x^2}(x_i)\right)\Delta x^2 + O(\Delta x^3) \\
&= a_i \frac{\partial \mathcal{W}}{\partial x}(x_i)\Delta x + \frac{\partial}{\partial x}\left(a\frac{\partial \mathcal{W}}{\partial x}\right)(x_i)\frac{\Delta x^2}{2} + O(\Delta x^3)
\end{aligned}$$

and the semi-discrete modified equation writes:

$$\frac{W_i^{n+1} - W_i^n}{\Delta t} = -\frac{\partial \mathcal{F}(\mathcal{W})}{\partial x}(x_i) + \frac{\Delta x}{2}\frac{\partial}{\partial x}\left(a\frac{\partial \mathcal{W}}{\partial x}\right)(x_i) + O(\Delta x^2).$$

Assuming moreover that:

$$W_i^{n+1} \approx W_i^n + \Delta t \frac{\partial \mathcal{W}}{\partial t}(t^n) + O(\Delta t^2),$$

we finally have the equivalent continuous equation:

$$\frac{\partial \mathcal{W}}{\partial t}(t^n, x_i) + \frac{\partial \mathcal{F}(\mathcal{W})}{\partial x}(t^n, x_i) = \frac{\Delta x}{2}\frac{\partial}{\partial x}\left(a\frac{\partial \mathcal{W}}{\partial x}\right)(t^n, x_i) + O(\Delta t) + O(\Delta x^2). \tag{C.1}$$

For the equation on the volume fraction, let us consider equation (4.36) of Chapter 4. One can show that the equivalent continuous equation reads:

$$\frac{\partial \alpha}{\partial t} + u\frac{\partial \alpha}{\partial x} = O(\Delta x^2) + O(\Delta t).$$

Then, the integration of the source term using an approximate method as presented previously, leads to the following modified equation with its first-order consistency error:

$$\frac{\partial \alpha}{\partial t} = \frac{p_1 - p_2}{\epsilon} + O\left(\frac{\Delta t}{\epsilon^2}\right). \tag{C.2}$$

Finally, the total equivalent continuous equation on α reads:

$$\frac{\partial \alpha}{\partial t} + u\frac{\partial \alpha}{\partial x} = \frac{p_1 - p_2}{\epsilon} + O\left(\frac{\Delta t}{\epsilon^2}\right) + O(\Delta x^2).$$

Now let us linearize the equations (C.1) and (C.2) about a constant and uniform state $\mathcal{W}^o$ such that: $\frac{\partial \mathcal{W}}{\partial x} = 0$, $u^o = 0$ and $p_1^o = p_2^o$. One can write: $\mathcal{W} = \mathcal{W}^o + \eta\mathcal{W}^1 + O(\eta^2)$, with $\eta \ll 1$, and the system of

equations can be written at first order in η:

$$\begin{cases} \dfrac{\partial \rho^1}{\partial t} + \rho^o \dfrac{\partial u^1}{\partial x} = a^o \dfrac{\Delta x}{2} \dfrac{\partial^2 \rho^1}{\partial x^2} \\ \dfrac{\partial Y^1}{\partial t} = a^o \dfrac{\Delta x}{2} \dfrac{\partial^2 Y^1}{\partial x^2} \\ \rho^o \dfrac{\partial u^1}{\partial t} + \dfrac{\partial p^1}{\partial x} = a^o \dfrac{\Delta x}{2} \rho^o \dfrac{\partial^2 u^1}{\partial x^2} \\ \dfrac{\partial \alpha^1}{\partial t} = \dfrac{p_1^1 - p_2^2}{\epsilon} \end{cases} \tag{C.3}$$

The first order perturbation is now assumed to be in the form of an acoustic wave, in the form: $\mathcal{W}^1 = \hat{\mathcal{W}} \exp(i\omega t - ikx)$. System (C.3) becomes then:

$$\begin{cases} \omega \hat{\rho} - \rho^o k \hat{u} = i a^o \dfrac{\Delta x}{2} k^2 \hat{\rho} \\ \omega \hat{Y} = i a^o k^2 \dfrac{\Delta x}{2} \hat{Y} \\ \rho^o \omega \hat{u} - k \hat{p} = i a^o \rho^o k^2 \dfrac{\Delta x}{2} \hat{u} \\ \omega \hat{\alpha} = -i \dfrac{\hat{p}_1 - \hat{p}_2}{\epsilon} \end{cases} \tag{C.4}$$

Let us also write $\hat{p}$ and $\hat{p}_1 - \hat{p}_2$ in terms of $\hat{\rho}$, $\hat{Y}$ and $\hat{\alpha}$:

$$\hat{p}_1 = A^o \hat{\rho} + B^o \hat{Y} + C^o \hat{\alpha}$$
$$\hat{p}_2 = D^o \hat{\rho} + E^o \hat{Y} + F^o \hat{\alpha}$$

with:

$$A^o = \frac{c_1^2 Y}{\alpha} \qquad B^o = \frac{c_1^2 \rho}{\alpha} \qquad C^o = -\frac{c_1^2 \rho Y}{\alpha^2}$$
$$D^o = \frac{c_2^2 (1-Y)}{1-\alpha} \qquad E^o = -\frac{c_2^2 \rho}{1-\alpha} \qquad F^o = \frac{c_2^2 \rho (1-Y)}{(1-\alpha)^2}$$

Then, one has:

$$\hat{p}_1 - \hat{p}_2 = (A^o - D^o)\,\hat{\rho} + (B^o - E^o)\,\hat{Y} + (C^o - F^o)\,\hat{\alpha}$$
$$\hat{p} = (\alpha^o A^o + (1-\alpha^o) D^o)\,\hat{\rho} + (\alpha^o B^o + (1-\alpha^o) E^o)\,\hat{Y} + (\alpha^o C^o + (1-\alpha^o) F^o)\,\hat{\alpha}$$

Now let us write the system (C.4) in its vectorial form:

$$\mathbb{A}(\rho^o, Y^o, \alpha^o, \omega, k) \left(\hat{\rho}, \hat{Y}, \hat{u}, \hat{\alpha}\right)^T = 0$$

This system has non trivial solutions, if matrix $\mathbb{A}$ is non-invertible, meaning:

$$\det(\mathbb{A}) = 0. \tag{C.5}$$

This last equality defines the dispersion relation for the first order-perturbation of system (C.3).

By developing (C.5) at first-order in $H = \frac{a^o \Delta x}{2}$, thus by assuming that $\Delta x \ll 1$, we obtain the following relation:

$$\begin{aligned}0 = & i\rho\omega^3 \left(C^o - F^o - i\epsilon\omega\right) \\ & + \rho\omega \left[3\omega H(C^o - F^o - i\epsilon\omega) + i(A^o F^o - C^o D^o + i\epsilon\omega(1-\alpha^o)D^o + i\epsilon\omega\alpha^o A^o)\right] k^2 \\ & + \rho H \left(A^o F^o - C^o D^o + i\epsilon\omega(1-\alpha^o)D^o + i\epsilon\omega\alpha^o A^o\right) k^4.\end{aligned}$$

If we introduce $L = C^o - F^o$, $M = A^o F^o - C^o D^o$, $N = (1-\alpha^o)D^o + \alpha^o A^o$ and $K = k^2$, the equation writes:

$$0 \quad = \quad i\omega^3 \left(L - i\epsilon\omega\right) \quad + \quad \omega \left[3\omega H(L - i\epsilon\omega) + i(M + i\epsilon\omega N)\right] K \quad + \quad H \left(M + i\epsilon\omega N\right) K^2 \quad \text{(C.6)}$$

The discriminant of this second-degree polynomial is:

$$\Delta = \omega^2 \left[3\omega H(L - i\epsilon\omega) + i(M + i\epsilon\omega N)\right]^2 - 4i\omega^3 H \left(L - i\epsilon\omega\right)\left(M + i\epsilon\omega N\right),$$

which writes, once more at first-order in H:

$$\begin{aligned}\Delta &= -\omega^2 (M + i\epsilon\omega N)^2 + 2i\omega^3 H \left(L - i\epsilon\omega\right)\left(M + i\epsilon\omega N\right) \\ &= -\omega^2 (M + i\epsilon\omega N)^2 \left(1 - 2i\omega H \frac{L - i\epsilon\omega}{M + i\epsilon\omega N}\right)\end{aligned}$$

and then in the limit $H \ll 1$:

$$\begin{aligned}\sqrt{\Delta} &\approx \pm i\omega(M + i\epsilon\omega N)\left(1 - i\omega H \frac{L - i\epsilon\omega}{M + i\epsilon\omega N} + O(H^2)\right), \\ &\approx \pm i\omega(M + i\epsilon\omega N) \pm \omega^2 H \left(L - i\epsilon\omega\right) + O(H^2).\end{aligned}$$

At first order in H, the two roots of polynomial (C.6) are thus:

$$\begin{aligned}K_+ &= \frac{-\omega\left[3\omega H(L - i\epsilon\omega) + i(M + i\epsilon\omega N)\right] + i\omega(M + i\epsilon\omega N) + \omega^2 H \left(L - i\epsilon\omega\right)}{2H \left(M + i\epsilon\omega N\right)} \\ &= \frac{-\omega^2 \left(L - i\epsilon\omega\right)}{M + i\epsilon\omega N}, \\ K_- &= \frac{-\omega\left[3\omega H(L - i\epsilon\omega) + i(M + i\epsilon\omega N)\right] - i\omega(M + i\epsilon\omega N) - \omega^2 H \left(L - i\epsilon\omega\right)}{2H \left(M + i\epsilon\omega N\right)} \\ &= \frac{-i\omega}{H} - \frac{2\omega^2 \left(L - i\epsilon\omega\right)}{M + i\epsilon\omega N}.\end{aligned}$$

The dispersion relation associated with K_+ is equivalent to the analytical dispersion relation of the 4-equation system, and in the limit $\epsilon \to 0$, one recovers Wood's sound velocity for the phase velocity, and no spatial attenuation.

K_- is associated with a second wave whose attenuation and phase velocity depend on $H = \frac{a^o \Delta x}{2}$. The smaller the parameter H the larger the wave atenuation. However, in the 3-equation system, a is of order ρc_{Wood}, while for the 4-equation system, it is ρc_{Frozen}: according to the value of α, c_{Frozen} can be two orders of magnitude larger than c_{Wood}. This second (and artificial) wave may disturb the 4-equation solution even for small values of ϵ.

Appendix D

Well-balanced schemes

As in Chapter 4, we consider a general system of equations, with barotropic pressure laws and with a gravity source term, that writes:

$$\left\{\begin{array}{rcccc} \partial_t(\rho\mathbf{u}) & + & \mathrm{div}_x\left(\rho\mathbf{u}\mathbf{u}^T + P\,\mathbf{Id}\right) & = & \rho\mathbf{g} \\ \partial_t\rho & + & \mathrm{div}_x(\rho\mathbf{u}) & = & 0 \\ \partial_t\mathbf{V} & + & \mathbf{u}\cdot\boldsymbol{\nabla}_x\mathbf{V} & = & \mathbf{0} \end{array}\right. \tag{D.1}$$

where we assume that the gravity acceleration applies only on the second directional component and is constant: $\mathbf{g} = (0, -g, 0)^T$ (in order to keep the same notations for 2D and 3D cases).

D.1 Issues of balance laws with source terms

For general systems of balance laws in the form:

$$\partial_t\mathbf{W} + \mathrm{div}\left(\mathbf{F}(\mathbf{W})\right) = \mathbf{S}(\mathbf{W}),$$

stationary solutions $\overline{\mathbf{W}}$ are defined by the equality:

$$\mathrm{div}\left(\mathbf{F}(\overline{\mathbf{W}})\right) = \mathbf{S}(\overline{\mathbf{W}}). \tag{D.2}$$

For bounded systems, with no particular boundary condition, this equilibrium can be obtained in very long times (comparing with the other characteristic times of the system) from initial solutions not too far from this equilibrium. However, usual numerical schemes are often inefficient in getting such a stationary state, due to large truncation errors near equilibrium, except at the price of a very fine discretization. In particular, Godunov-type schemes with operator splitting generate small waves that prevent a discrete equilibrium to be set.

Over the past years, many works have contributed to develop so-called *well-balanced* schemes, i. e. schemes that are able to preserve a stationary initial solution. In Cargo and le Roux (1994), Euler equations with a gravity source term is addressed, where the use of the potential of the gravity field leads to an homogeneous system of equations and the well-balanced property. Then in Greenberg and Roux (1994); Gosse and Roux (1996), other source terms are studied for scalar laws and are reformulated as non-conservative products. Despite the restrictions on the form of the source term, and on the properties of the flux function, this method was the first version of the current approaches for the shallow-water equations with a topography source term Gosse (2000); Chinnayya, Roux, and Seguin (2004); Audusse, Bouchut, Bristeau, Klein, and Perthame (2004), and is also widely used for atmospheric simulations.

An other method consists in using the *consistency in the integral sense*, as developed in Gallice (2002); Gallice (2003). This consists in writing generalized jump relations for the Riemann's problem, that takes into account a term consistent with the source term. Numerical fluxes are then determined accounting for the sources effects and lead to well-balanced schemes. This method was later used for asymptotic-preserving schemes as in Chalons, Girardin, and Kokh (2013).

D.2 A method based on Cargo-Leroux equations

D.2.1 A new system of equations

Following the lines of Cargo and le Roux (1994); Greenberg and Roux (1994), we consider the resolution of system (D.1) in the y-direction. There is no modification in the other directions, and a directional splitting is compatible with the method. Let us introduce the potential variable q such that:

$$\begin{array}{rcl} \partial_y q &=& -\rho g \\ \partial_t q &=& \rho g v \end{array}. \tag{D.3}$$

One can show that this potential exists by noting that the conservation equation on ρ, multiplied by the constant g can be interpreted as the equation on the divergence of the vector $(\rho g, \rho g v)^T$ (with derivatives with respect to time t and direction y). From the definition of q and the equation on ρ, it is easy to deduce the equation on q:

$$\partial_t q + v \partial_y q = 0$$

Let us then rewrite system (D.1) in 1D with the new variables q and $\phi = P - q$:

$$\left\{ \begin{array}{ccccc} \partial_t (\rho v) & + & \partial_y \left(\rho v^2 + \phi\right) & = & 0 \\ \partial_t \rho & + & \partial_y (\rho v) & = & 0 \\ \partial_t q & + & v \partial_y q & = & 0 \\ \partial_t \mathbf{V} & + & v \partial_y \mathbf{V} & = & \mathbf{0} \end{array} \right. \tag{D.4}$$

D.2.2 Hydrostatic equilibrium

The continuous hydrostatic equilibrium is defined by:

$$\begin{array}{l} v = 0 \\ \partial_y P = -\rho g \end{array} \tag{D.5}$$

and one can compute the continuous hydrostatic equilibrium by integrating the second equation. Using the pressure law, we can note that this equation comes down to solve a first order ODE that writes:

$$c^2(\rho, \mathbf{V}) \partial_y \rho = -\rho g - d(\rho, \mathbf{V})\, P \partial_y \mathbf{V} \tag{D.6}$$

where we assume that the profile of $\mathbf{V}$ is given and $d(\rho, \mathbf{V}) = \partial_{\mathbf{V}} P$. According to the chosen thermodynamic laws, this ODE may be impossible to solve explicitly.

Let us now study a discrete hydrostatic equilibrium. Let us assume that the y-direction is discretized, and call $(y_{i+\frac{1}{2}})_i$ the points of this discretization defining cells, whose centers are located at $(y_i)_i$. In a classic finite volume approach, variables are assumed constant over cells, leading to discontinuities at the cells interfaces (in $y_{i+\frac{1}{2}}$). Here, let us make the assumption that the pressure profile is differentiable, so as to integrate the ODE (D.5) properly. Then, from cell $\mathcal{K}_i$ to its neighbor, one will have:

$$\begin{aligned} \int_{y_i}^{y_{i+1}} \partial_y P &= -\int_{y_i}^{y_{i+1}} \rho g dy \\ P(y_{i+1}) - P(y_i) &= -\int_{y_i}^{y_{i+1}} \rho g dy \\ &= -\rho(y_i) g \left(y_{i+\frac{1}{2}} - y_i\right) - \rho(y_{i+1}) g \left(y_{i+1} - y_{i+\frac{1}{2}}\right) \end{aligned}$$

and we can define cell by cell a discrete hydrostatic equilibrium, which is an approximation at first order of the continuous one. Let us note also that, knowing $P(y_i)$, the value of $P(y_{i+1})$ in the example is then given implicitly, as we have $\rho(y_{i+1}) = f\left(P(y_{i+1}), \mathbf{V}(y_{i+1})\right)$ where the function $f : (P, \mathbf{V}) \mapsto \rho$ depends on the state laws. The computation of this discrete equilibrium may then be costly.

The final discrete equilibrium we consider writes:

$$\begin{aligned} v_i &= 0 \\ P_i &= P_0 \quad -g \sum_{0 \leqslant j < i} \left[\rho_j \frac{\Delta y_j}{2} + \rho_{j+1} \frac{\Delta y_{j+1}}{2} \right] \end{aligned} \tag{D.7}$$

where we use the notations $P_i = P(y_i)$, $\rho_i = \rho(y_i)$, and $\Delta y_i = y_{i+\frac{1}{2}} - y_{i-\frac{1}{2}}$.

In the following, we are designing a scheme that preserves this discrete equilibrium, when it is given as an initial condition.

Remark 41. *There are many possible discrete equilibria. Finer approximations could be found by using a Taylor development of the exact solution of ODE (D.5), and then take higher order terms for expressing the density ρ in one cell. For instance, if the sound velocity c is a constant of ρ, it is then easy to solve equation (D.5) and use this solution for building the hydrostatic profile.*

Suliciu's flux and new source term Let us use Suliciu's relaxation method applied to system (D.4). Let us first note that the variable ϕ satisfies the following equation:

$$\partial_t \phi + v \partial_y \phi + \rho c^2(\rho, \mathbf{V}) \partial_y v = 0.$$

Let us introduce the relaxation variable Φ, that satisfies the following equation:

$$\partial_t \Phi + v \partial_y \Phi + \frac{a^2}{\rho} \partial_y v = \lambda(\phi - \Phi),$$

where a is a constant defined locally as: $a = \rho c(\rho, \mathbf{V})$, and the relaxation system now writes:

$$\left\{ \begin{array}{ccccc} \partial_t (\rho v) & + & \partial_y \left(\rho v^2 + \Phi \right) & = & 0 \\ \partial_t (\rho \Phi) & + & \partial_y \left(\rho \Phi v + a^2 v \right) & = & \lambda \rho (\phi - \Phi) \\ \partial_t \rho & + & \partial_y (\rho v) & = & 0 \\ \partial_t q & + & v \partial_y q & = & 0 \\ \partial_t \mathbf{V} & + & v \partial_y \mathbf{V} & = & \mathbf{0} \end{array} \right. \tag{D.8}$$

which can be solved using the same formulae as in Chapter 4. Let us then recall that the intermediate states for v and Φ write in function of the left and right states of the Riemann's problem:

$$\begin{aligned} v^* &= \frac{v_L + v_R}{2} + \frac{\Phi_L - \Phi_R}{2a} \\ \Phi^* &= \frac{\Phi_L + \Phi_R}{2} + a \frac{v_L - v_R}{2} \end{aligned}$$

and the numerical fluxes can be given by the physical flux estimated with the solution of the Riemann's problem:

$$\mathbf{F}_{i+\frac{1}{2}} = \mathbf{F}\left(\mathcal{W}\left(0, \mathbf{W}_i, \mathbf{W}_{i+1}\right) \right) \tag{D.9}$$

Then, as for Suliciu's relaxation method, only the variables ρ, v and $\mathbf{V}$ are updated using the fluxes. The auxiliary variables q and Φ must be initialized using the thermodynamic and potential laws.

The question is now to properly define Φ, meaning, to define ϕ and q, for the scheme to preserve the hydrostatic equilibrium defined by (D.7).

Proposition 6. Let the discrete value of q be defined as:

$$q_i = -\int_{y_0}^{y_i} \rho g dy \tag{D.10}$$

then the fluxes (D.9) corresponding to the system (D.8) preserve an hydrostatic equilibrium.

Proof. Let us assume that initially, the hydrostatic equilibrium (D.7) is satisfied in the whole domain, and especially in cell $\mathcal{K}_i$. To preserve the hydrostatic equilibrium, the fluxes of system (D.8) at the interfaces must be null or counterbalance. Two features are then to be verified:

1. the intermediate velocity must be null, so that fluxes on ρ and $\mathbf{V}$ are also null,
2. the two interfacial fluxes $\mathbf{F}_{i-\frac{1}{2}}$ and $\mathbf{F}_{i+\frac{1}{2}}$ for momentum must be equal.

Let us consider the solution of the Riemann's problem at interface $\mathcal{I}_{i+\frac{1}{2}}$. We already know that $v_i = 0$ and $v_{i+1} = 0$. What about v^*?

We have:

$$\begin{aligned} v^* &= \frac{\Phi_i - \Phi_{i+1}}{2a} \\ &= \frac{1}{2a}\left(P_i - q_i - P_{i+1} + q_{i+1}\right) \\ &= \frac{1}{2a}\left(P_i - P_{i+1} - \int_{y_i}^{y_{i+1}} \rho g dy\right) \end{aligned}$$

and according to the definition of the discrete equilibrium (D.7), one has $v^* = 0$. The solution to the Riemann's problem for velocity is then null and the first condition the scheme should verify is satisfied.

Let us now compute the difference of fluxes $\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}}$ for momentum to check the second condition. We can first note that because $v_i = 0$, the solution to the Riemann's problem is the intermediate state. The flux of momentum is then $\mathbf{F}^* = \rho^* {u^*}^2 + \Phi^*$. Now, we have seen previously that $u^* = 0$, then it remains to compute $\mathbf{F}_{i+\frac{1}{2}} - \mathbf{F}_{i-\frac{1}{2}} = \Phi_{i+\frac{1}{2}} - \Phi_{i-\frac{1}{2}}$.

By definition, we have:

$$\begin{aligned} \Phi_i &= P_i - q_i \\ &= P_i + \int_{y_0}^{y_i} \rho g dy \end{aligned}$$

and

$$\begin{aligned} \Phi_{i+1} &= P_{i+1} - q_{i+1} \\ &= P_{i+1} + \int_{y_0}^{y_{i+1}} \rho g dy \\ &= P_{i+1} + \int_{y_0}^{y_i} \rho g dy + \int_{y_i}^{y_{i+1}} \rho g dy \end{aligned}$$

As the initial pressure profile is at equilibrium, the last line of the second inequality can be written as:

$$\begin{aligned} \Phi_{i+1} &= P_i + \int_{y_0}^{y_i} \rho g dy \\ &= \Phi_i \end{aligned}$$

and finally $\Phi_{i+\frac{1}{2}} = \Phi_i$. In the same way, one has:

$$\begin{aligned}\Phi_{i-1} &= P_{i-1} - q_{i-1} \\ &= P_{i-1} + \int_{y_0}^{y_{i-1}} \rho g dy \\ &= P_{i-1} + \int_{y_0}^{y_i} \rho g dy + \int_{y_i}^{y_{i-1}} \rho g dy \\ &= P_i + \int_{y_0}^{y_i} \rho g dy \\ &= \Phi_i\end{aligned}$$

and $\Phi_{i+\frac{1}{2}} = \Phi_i = \Phi_{i-\frac{1}{2}}$. The fluxes of momentum are then equal and the equilibrium is preserved. □

The definition of q_i as given by (D.10) is not practical, as it necessitates an integration on a whole vertical column of fluid. We are presenting below a modification of the scheme that makes it simpler to use. Let us introduce $Q_{i+\frac{1}{2}}$, defined by:

$$Q_{i+\frac{1}{2}} = -\int_{y_i}^{y_{i+1}} \rho g dx = q_{i+1} - q_i$$

and note that:

$$\Phi_{i+1} - \Phi_i = P_{i+1} - P_i - Q_{i+\frac{1}{2}}$$

and

$$\Phi_{i+1} + \Phi_i = P_{i+1} + P_i + 2q_i - Q_{i+\frac{1}{2}}.$$

However, whatever the solution of the Riemann's problem at interfaces $\mathcal{I}_{i-\frac{1}{2}}$ and $\mathcal{I}_{i+\frac{1}{2}}$, we can show that the contribution of Φ in the numerical flux writes:

$$\Phi^*_{i\pm\frac{1}{2}} = P^*_{i\pm\frac{1}{2}} \pm \beta Q_{i\pm\frac{1}{2}} - q_i \tag{D.11}$$

with $\beta = 0, -\frac{1}{2}, -1$ according to the states of the Riemann's problem. Then, the contribution of q_i will cancel when subtracting the fluxes. As it has no role in the rest of the Riemann's problem, we can ignore it, when updating cell $\mathcal{K}_i$, and then get rid off its costly initialization.

Let us then give new definitions of the intermediate states of the Riemann's problem, when considering cell $\mathcal{K}_i$, given by:

$$\begin{aligned}\hat{v}_{i\pm\frac{1}{2}} &= \frac{v_i + v_{i\pm1}}{2} \pm \frac{P_i - P_{i\pm1} \pm Q_{i\pm\frac{1}{2}}}{2a} \\ \hat{P}_{i\pm\frac{1}{2}} &= \frac{P_i + P_{i\pm1} \mp Q_{i\pm\frac{1}{2}}}{2} \pm a\frac{v_i - v_{i\pm1}}{2}\end{aligned} \tag{D.12}$$

One must be aware that these formulae are valid for cell $\mathcal{K}_i$ only! They will differ in cells $\mathcal{K}_{i-1}$ and $\mathcal{K}_{i+1}$, leading to interfacial fluxes that are not equal from one side of the face to the other: the scheme is not conservative, but it was expected, as in reality, the equations contain a source term.

D.2.3 Boundary conditions:

For vertical fluxes, that are computed at the border of the domain, one must also ensure that an initial equilibrium is preserved.

If we consider reflective boundary conditions, for which the matter fluxes are null, one can enforces the boundary flux for cell $\mathcal{K}_i$ to be:

$$\mathbf{F}(\mathbf{W}_i) = (\phi_i, 0, \mathbf{0})^T$$

where $\mathbf{W} = (\rho v, \rho, \rho \mathbf{V})^T$.

In the case of an open boundary condition, we must impose that the flux at the border balances the flux which is computed at the opposite face of the cell. We can then just copy it.

Remark 42. *One can use layers of ghost cells so as to compute the boundary conditions. In these cases, the values that are set in these ghost cells must satisfy the conditions on the fluxes presented just above.*

D.3 Numerical features and AMR

D.3.1 Non-conforming meshes:

The scheme developed in the previous paragraph applies on 1D-meshes. It extends without difficulty on multidimensional cartesian grids, by using the directional splitting. Let us now see how it can be used on cartesian non-conforming meshes.

Let us first verify that in the y-direction, Proposition 6 is true, and that we can use the modified intermediate states (D.12).

Let us first define an hydrostatic equilibrium in a case as in Figure D.1a. In fact, by projecting on the y-axis, one can use directly formula (D.7) to define the hydrostatic equilibrium. Let us now check the conditions to preserve the equilibrium:

- in the Riemann's problems at the interfaces, the intermediate value of velocity is null, as one can follow the lines of the proof of Proposition 6.
- the sum of the fluxes for momentum must be null, in the sense (following the notations of the Figure):

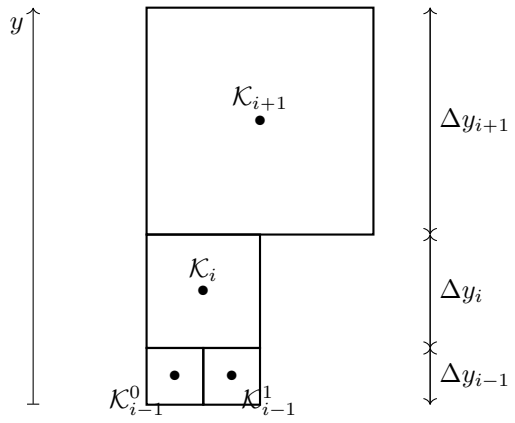
$$\frac{\Phi^0_{i-\frac{1}{2}}}{2\Delta y_i} + \frac{\Phi^1_{i-\frac{1}{2}}}{2\Delta y_i} - \frac{\Phi_{i+\frac{1}{2}}}{\Delta y_i} = 0. \tag{D.13}$$

  According to the proof of proposition 6, one has: $\Phi^0_{i-\frac{1}{2}} = \Phi_i$, $\Phi^1_{i-\frac{1}{2}} = \Phi_i$ and $\Phi_{i+\frac{1}{2}} = \Phi_i$, and then the residual for momentum is indeed null.

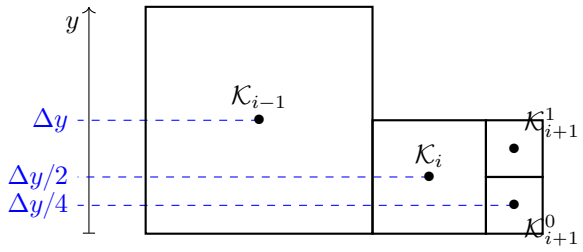
The discrete hydrostatic equilibrium is thus preserved in the configuration of Figure D.1a. One could show that this is also the case for any other configuration.

We now wonder whether the modified values of intermediate states (D.12) can be used, when computing the fluxes in cell $\mathcal{K}_i$. In fact, when using the residual formulation (D.13), one can see that the contribution in q_i from the formula (D.11) also cancels.

We can then conclude that the scheme is compatible in the y-direction for a non-conforming mesh.



(a) Case of non-conforming mesh



(b) Case of non-conforming mesh

Then, let us study the other directions. We consider x-direction (z-direction follows the same lines). In an AMR case (see an example in Figure D.1b), the centers of neighboring cells in the x-direction can be at differing heights. Then, according to the hydrostatic equilibrium profile, the pressure within these cells can differ, and an horizontal velocity is generated, if no particular treatment is done to preserve the equilibrium.

An idea of scheme modification is to subtract the gravity contribution in the pressure difference between the cells, in the following way: let us assume that we are in the configuration of Figure D.1b and we are computing the fluxes between cells $\mathcal{K}_i$ and $\mathcal{K}^0_{i+1}$. Assuming that the hydrostatic equilibrium profile is set initially, and the reference height is given at y_0 as in Figure D.1b, then one will have:

$$P(y^0_{i+1}) = P_0 - \int_{y_0}^{y_0+\Delta y/4} \rho_{i+1}(y) g dy$$

and

$$\begin{aligned} P(y_i) &= P_0 - \int_{y_0}^{y_0+\Delta y/2} \rho_i(y) g dy \\ &= P_0 - \int_{y_0}^{y_0+\Delta y/4} \rho_i(y) g dy - \int_{y_0+\Delta y/4}^{y_0+\Delta y/2} \rho_i(y) g dy \\ &= P(y^0_i) - \int_{y_0+\Delta y/4}^{y_0+\Delta y/2} \rho_i(y) g dy \end{aligned}$$

where the pressure $P(y^0_i)$ is evaluated at the same height as pressure $P(y^0_{i+1})$. We will then solve the Riemann's problem at the interface between $\mathcal{K}_i$ and $\mathcal{K}^0_{i+1}$ by using on the left side the value $P(y^0_i)$ instead of $P(y_i)$.

Here is the main defect of the method, and in particular of the chosen discrete hydrostatic profile: even at equilibrium, there is no reason for $P(y^0_i)$ to be exactly equal to $P(y^0_{i+1})$. Indeed, one considers that the density ρ is constant in each cell, but this is only an approximation at first order of the ODE (D.5). To see that, let us assume that $\rho_i(y_0) = \rho_{i+1}(y_0) = f(P_0, \mathbf{V}_0)$. Let us assume that $\mathbf{V}(y_i) = \mathbf{V}(y^0_{i+1}$, then function f is the same in cells $\mathcal{K}_i$ and $\mathcal{K}^0_{i+1}$ and:

$$P^0_{i+1} = P_0 - f(P^0_{i+1}) g \frac{\Delta y}{4},$$

$$\begin{aligned} P^0_i &= P_i + f(P_i) g \frac{\Delta y}{4} \\ &= P_0 - f(P_i) g \frac{\Delta y}{2} + f(P_i) g \frac{\Delta y}{4} \\ &= P_0 - f(P_i) g \frac{\Delta y}{4}. \end{aligned}$$

The two formulations above are different, as the one for P^0_{i+1} is implicit, while the expression of P^0_i is given explicitly (while P_i is computed implicitly!).

Remark 43. *As noted previously, if the sound velocity c is independent of ρ, we can use the exact solution of ODE (D.5), which is in exponential form, and an appropriate estimation of q enables to maintain the hydrostatic equilibrium.*

We however retain the following strategy for computing the fluxes in the x-direction between cells $\mathcal{K}_i$ and $\mathcal{K}_{i+1}$:

- if cell $\mathcal{K}_{i+1}$ is the same size as cell $\mathcal{K}_i$, then we apply the usual solver with states $\mathbf{W}_i$ and $\mathbf{W}_{i+1}$,
- if $\mathcal{K}_{i+1}$ is smaller than cell $\mathcal{K}_i$ and their centers are not aligned horizontally, let y_{i+1} be the y-coordinate of the center of cell $\mathcal{K}_{i+1}$, then we compute the flux using variables $\mathbf{W}_{i+1}$ and $\tilde{\mathbf{W}}_i$

with:

$$\begin{aligned}\tilde{u}_i &= u_i \\ \tilde{P}_i &= P_i - \rho_i g\left(y_{i+1} - y_i\right) \\ \tilde{\rho}_i &= f\left(\tilde{P}_i\right) \\ \tilde{\mathbf{V}}_i &= \mathbf{V}_i\end{aligned}$$

- if $\mathcal{K}_{i+1}$ is greater than cell $\mathcal{K}_i$, let y_i be the y-coordinate of the center of cell $\mathcal{K}_i$, then we compute the flux using variables $\tilde{\mathbf{W}}_{i+1}$ and $\mathbf{W}_i$ with:

$$\begin{aligned}\tilde{u}_{i+1} &= u_{i+1} \\ \tilde{P}_{i+1} &= P_{i+1} - \rho_i g\left(y_i - y_{i+1}\right) \\ \tilde{\rho}_{i+1} &= f\left(\tilde{P}_{i+1}\right) \\ \tilde{\mathbf{V}}_{i+1} &= \mathbf{V}_{i+1}\end{aligned}$$

We will see with the numerical results, that this approximation gives better results than the non-modified schemes, and that it is possible to define a discrete hydrostatic equilibrium, based on a spatial discretization at the scale of the smallest cells, that is preserved.

Remark 44. *This correction looks like a second-order reconstruction, with the slope defined by the hydrostatic gradient. When considering that the larger cell is split into two smaller ones, one can note that the density rho is not conserved. When reconstructing the pressure profile, we must however ensure that density remains positive.*

D.3.2 Second order reconstruction:

We consider here the extension of the well-balanced scheme to the MUSCL-Hancock second-order scheme.

Let us first study the $y-$direction. We make the following choice for the reconstructed variables: $\mathbf{U} = (v, \phi, P, \mathbf{V})^T$, where v is the $y-$component of the velocity $\mathbf{u}$. Let us again consider the configuration of Figure D.1a and have a look at the slope between cell $\mathcal{K}_i$ and $\mathcal{K}_{i+1}$. The gradient for variable ϕ, is given by:

$$\begin{aligned}\sigma_\phi^{i+\frac{1}{2}} &= 2\frac{\phi_{i+1} - \phi_i}{\Delta y_{i+1} + \Delta y_i} \\ &= 2\frac{P_{i+1} - q_{i+1} - P_i + q_i}{\Delta y_{i+1} + \Delta y_i} \\ &= 2\frac{P_{i+1} - P_i + \int_{y_i}^{y_{i+1}} \rho g dy}{\Delta y_{i+1} + \Delta y_i}\end{aligned}$$

Assuming that initially, the discrete hydrostatic equilibrium is set in the domain, one has $\sigma_\phi^{i+\frac{1}{2}} = 0$. Similarly, one will have $\sigma_\phi^{i-\frac{1}{2},0} = 0$ and $\sigma_\phi^{i-\frac{1}{2},1} = 0$. If $v = 0$ initially, then using the same definition of the auxiliary variable as previously leads to the same formulae as for the first-order scheme: $v^* = 0$, meaning that all matter fluxes are null, and the residual for the vertical momentum is null too (D.13).

Remark 45. *For an initial condition that is not at equilibrium, we have chosen to reconstruct the pressure variable P as a primitive variable, to then estimate the density ρ at second order in the Riemann's problem. While ϕ can be considered as a pressure deprived from its hydrostatic effects, these effects must nevertheless be accounted for in the other variables.*

In $x-$direction, one has two possibilities:

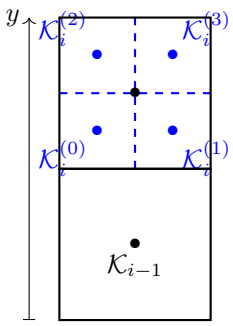


Figure D.1: *Refining / coarsening sketch.*

1. we could ignore the effects of gravity, and use a simple reconstruction on $\mathbf{U} = (u, P, \mathbf{V})^T$,
2. for configurations such in Figure D.1b we can first use an hydrostatic profile reconstruction in the largest cells to get more accurate values in the virtual smaller cells and then use this first reconstruction to compute the gradients of $\mathbf{U} = (u, P, \mathbf{V})^T$.

With the first possibility and in the case of an initial hydrostatic equilibrium, one will have $P(y^0_{i+1}) < P(_i) < P(^1_{i+1})$, and with the minmod slope limiters, the final slope for pressure will be automatically be null. This degeneracy to a first order scheme could be annoying for configurations close to equilibrium, but where some acoustic effects necessitate a fine resolution.

With the second option, depending on the discrete hydrostatic equilibrium set initially, the scheme is almost well-balanced, as we have seen for the first-order version. In non-equilibrium cases, the effects of gravity will have moreover less effects on the horizontal pressure gradients.

D.3.3 Refining and coarsening cells:

If for any reason, cells have to be coarsened of refined, in order to maintain an hydrostatic equilibrium, some specific treatment has to be done.

Let us first consider the case of coarsening, and refer to the configuration of Figure D.1. Due to the conservation of mass, one will have $\rho_i = \frac{1}{4}\sum_k \rho_i^{(k)}$. By this average, the hydrostatic equilibrium will not be preserved.

D.4 Two-fluid equilibrium model

Let us end this part by studying a two-phase flow described by the 3-equation system (from Chapter 1 and Chapter 4), that writes with the gravity source term:

$$\left\{\begin{array}{ccccc} \partial_t \rho & + & \mathrm{div}\,(\rho \mathbf{u}) & = & 0 \\ \partial_t (\rho Y) & + & \mathrm{div}\,(\rho Y \mathbf{u}) & = & 0 \\ \partial_t (\rho \mathbf{u}) & + & \mathrm{div}\left(\rho \mathbf{u}\mathbf{u}^T + p\,\mathbf{Id}\right) & = & \rho \mathbf{g} \end{array}\right. \tag{D.14}$$

and the pressure is defined by the equality $p = p_1(\rho, Y) = p_2(\rho, Y)$, where $p_1 : (\rho, Y) \mapsto p_1(\rho, Y)$ and $p_2 : (\rho, Y) \mapsto p_2(\rho, Y)$ are the pressure laws for fluids 1 and 2 respectively. In this case, the variable $\mathbf{V}$ may be associated with the mass fraction Y in order to apply the relations from the previous paragraphs.

D.4.1 Hydrostatic equilibrium:

First, we can study the ODE (D.5) and show that its exact resolution is out of reach in most cases. Indeed, let us assume that the profile of mass fraction is given: $Y(y)$, then, expressing ρ as a function of

p and Y gives:

$$\rho(p,Y) = \frac{\rho_1(p)\rho_2(p)}{Y\rho_2(p) + (1-Y)\rho_1(p)} \tag{D.15}$$

with $\rho_1 : p \mapsto \rho_1(p)$ and $\rho_2 : p \mapsto \rho_2(p)$ are given by barotropic state laws, and $\rho_1'(p) = c_1^{-2}$, $\rho_2'(p) = c_2^{-2}$. Let us moreover assume that c_1 and c_2 are constant (linearized pressure laws). Equation (D.5) can now be written:

$$\left(\frac{Y(y)}{\rho_1(p)} + \frac{1-Y(y)}{\rho_2(p)}\right)\partial_y p = -g.$$

This ODE is hardly integrable in the general case. Let us just study the case when the profile of Y is uniform: $Y(y) = \overline{Y}$. Then, the solution, integrated from $y_1 = 0$ to $y_2 = y$ is given by:

$$\begin{aligned}
\overline{Y}c_1^2\left[\ln(\rho_1(p(y))) - \ln(\rho_1(p(0)))\right] + (1-\overline{Y})c_2^2\left[\ln(\rho_2(p(y))) - \ln(\rho_2(p(0)))\right] &= -gy \\
\ln\left[\left(\frac{\rho_1(p(y))}{\rho_1(p(0))}\right)^{\overline{Y}c_1^2}\left(\frac{\rho_2(p(y))}{\rho_2(p(0))}\right)^{(1-\overline{Y})c_2^2}\right] &= -gy \\
\rho_1(p(y))^{\overline{Y}c_1^2}\rho_2(p(y))^{(1-\overline{Y})c_2^2} &= \rho_1(p(0))^{\overline{Y}c_1^2}\rho_2(p(0))^{(1-\overline{Y})c_2^2}\exp\left(-gy\right)
\end{aligned}$$

and this relation is again an implicit function of p. When $\overline{Y} = 0$, this relation becomes:

$$\rho_2(p(y)) = \rho_2(p(0))\exp\left(-\frac{gy}{c_2^2}\right),$$

and when $\overline{Y} = 1$ it is:

$$\rho_1(p(y)) = \rho_1(p(0))\exp\left(-\frac{gy}{c_1^2}\right).$$

However, in two-phase flows, the profile of Y may take any form, and the exact hydrostatic profile is not known.

Let us then consider the discrete profile, such as given by formula (D.7). This profile can be used in a two-phase flow, with the expression for $f(P,\mathbf{V})$ given by (D.15) ($P = p$ and $\mathbf{V} = Y$).

D.4.2 Vertical and horizontal equilibria:

In the previous paragraph, we have chosen the mass fraction Y as the phasic indicator, and we have defined the initial profiles using this variable. Another possibility would have been to choose the volume fraction α, which, in the 3-equation system, is a variable defined by the equilibrium of pressures. Using:

$$\rho(p,\alpha) = \alpha\rho_1(p) + (1-\alpha)\rho_2(p)$$

we could then write the ODE (D.5) as:

$$\partial_y p = -\left(\alpha\rho_1(p) + (1-\alpha)\rho_2(p)\right)g$$

One can note that in the vertical direction, any profile of Y (or α) can be used to define hydrostatic equilibria. We could then have an air pocket lying under a water layer. This configuration is possible because there is no difference in the velocities of the two fluids, preventing bubbles from going upwards in a liquid phase.

While the vertical profile of Y can have any form, the horizontal one must be uniform. Indeed, an horizontal gradient of Y would create a gradient of density ρ and then on pressure. As there is no source term in this direction to counterbalance the gradient of p, then motion appears. This configuration is typically the case of the dam breaking problem that we will study later in this thesis.

Appendix E

Tutorial on using Gmsh for creating macro-meshes

E.1 Purpose

We would like to create geometries a bit more complex than only squares and cubes, and do simulation on them with CanoP. In consequence of these complex geometries, a macro-mesh must also be defined.

Up to know, one branch of CanoP is able to deal with macro-cells that have been read externally through an ABAQUS-type input file.

Some cleaning of CanoP has remains to be done, however, as just order 1 cases have been computed, and only reflective boundary conditions have been corrected !

E.2 2D - tutorial

E.2.1 Creating geometry and subdomains

We here present the main steps to create a `*.inp` file, using Gmsh , that will be read by CanoP. We want to represent the figure given below:

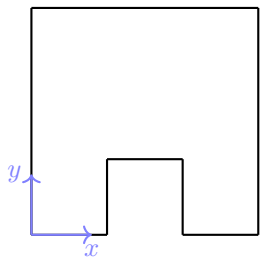


Figure E.1: *Geometry of the domain.*

First, we are going to subdivide this figure into 5 rectangular main subdomains such that this first macro-mesh is conforming :

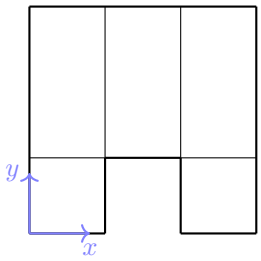


Figure E.2: *Conforming mesh.*

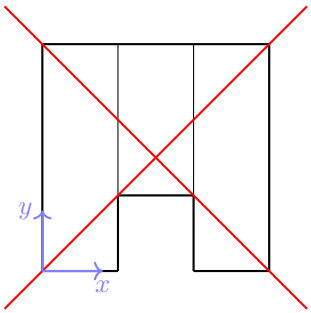


Figure E.3: *Non-conforming mesh.*

To do so, we first place all the points (for our geometry, there must be 12 points), using :

```
> Modules
      > Geometry
          > Add
              > Point
```

We then draw all the lines by linking the necessary points and using :

```
> Modules
      > Geometry
          > Add
              > Straight line
```

And finally, we define the surface, by selecting the lines (in counterclockwise order) and using :

```
> Modules
      > Geometry
          > Add
              > Plane surface
```

To see the corresponding `*.geo` file, one may click on :

```
> Modules
      > Geometry
          > Edit file
```

and one may find something like :

```
// Gmsh project created on Fri Nov  6 13:26:53 2015
Point(1) = {0, 0, 0, 1.0};
Point(2) = {0, 1, 0, 1.0};
Point(3) = {1, 0, 0, 1.0};
Point(4) = {2, 0, 0, 1.0};
Point(5) = {3, 0, 0, 1.0};
Point(6) = {3, 1, 0, 1.0};
Point(7) = {3, 3, 0, 1.0};
Point(8) = {2, 1, 0, 1.0};
Point(9) = {2, 3, 0, 1.0};
Point(10) = {1, 3, 0, 1.0};
Point(11) = {1, 1, 0, 1.0};
```

```
Point(12) = {0, 3, 0, 1.0};
Line(1) = {1, 2};
Line(2) = {2, 12};
Line(3) = {3, 11};
Line(4) = {11, 10};
Line(5) = {4, 8};
Line(6) = {8, 9};
Line(7) = {5, 6};
Line(8) = {6, 7};
Line(9) = {1, 3};
Line(10) = {4, 5};
Line(11) = {2, 11};
Line(12) = {11, 8};
Line(13) = {8, 6};
Line(14) = {12, 10};
Line(15) = {10, 9};
Line(16) = {9, 7};
Line Loop(17) = {9, 3, -11, -1};
Plane Surface(18) = {17};
Line Loop(19) = {2, 14, -4, -11};
Plane Surface(20) = {19};
Line Loop(21) = {12, 6, -15, -4};
Plane Surface(22) = {21};
Line Loop(23) = {13, 8, -16, -6};
Plane Surface(24) = {23};
Line Loop(25) = {10, 7, -13, -5};
Plane Surface(26) = {25};
```

If some modification has to be applied in this text file, on has to save them and then reload into Gmsh :

```
> Modules
      > Geometry
          > Reload
```

E.2.2 Creating the macro-cells

Now we have defined the geometry and the main subdomains, we can generate a mesh using Gmsh . To do a cartesian one in each subdomain, one must configure do three operations :

- defining the number of nodes per line, using :

  ```
  > Modules
     > Mesh
         > Define
             > Transfinite
                 > Line
  ```

  and must be aware that for defining 2 cells, one needs 3 nodes...

- imposing that surfaces must be meshed using structured mesh:

  ```
  > Modules
     > Mesh
         > Define
             > Transfinite
                 > Surface
  ```

- “recombining" the surfaces:

```
> Modules
    > Mesh
        > Define
            > Recombine
```

New lines are then added to the `*.geo` file :

```
Transfinite Line {1, 9, 11, 3, 12, 13, 5, 7, 10, 14, 15, 16} = 3 Using Progression 1;
Transfinite Line {2, 4, 6, 8} = 5 Using Progression 1;
Transfinite Surface {18};
Transfinite Surface {20};
Transfinite Surface {22};
Transfinite Surface {24};
Transfinite Surface {26};
Recombine Surface {18, 20, 22, 24, 26};
```

and the mesh looks like :

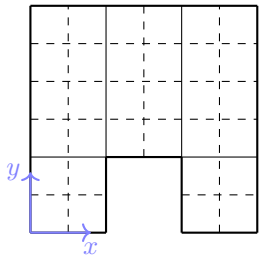


Figure E.4: *Macro-mesh.*

E.2.3 Saving as ABAQUS file

The last step is to save the mesh as an `ABAQUS`-type file (by selecting the `*.inp` extension). The file that is saved looks like :

```
*Heading
example-tuto.inp
*NODE
1, 0, 0, 0
2, 0, 1, 0
3, 1, 0, 0
4, 2, 0, 0
5, 3, 0, 0
6, 3, 1, 0
7, 3, 3, 0
8, 2, 1, 0
9, 2, 3, 0
10, 1, 3, 0
11, 1, 1, 0
12, 0, 3, 0
13, 0, 0.5, 0
14, 0, 1.5, 0
15, 0, 2.0, 0
16, 0, 2.5, 0
17, 1, 0.5, 0
```

```
18, 1, 1.5, 0
19, 1, 2.0, 0
20, 1, 2.5, 0
21, 2, 0.5, 0
22, 2, 1.5, 0
23, 2, 2.0, 0
24, 2, 2.5, 0
25, 3, 0.5, 0
26, 3, 1.5, 0
27, 3, 2.0, 0
28, 3, 2.5, 0
29, 0.5, 0, 0
30, 2.5, 0, 0
31, 0.5, 1, 0
32, 1.5, 1, 0
33, 2.5, 1, 0
34, 0.5, 3, 0
35, 1.5, 3, 0
36, 2.5, 3, 0
37, 0.5, 0.5, 0
38, 0.5, 1.5, 0
39, 0.5, 2.0, 0
40, 0.5, 2.5, 0
41, 1.5, 1.5, 0
42, 1.5, 2.0, 0
43, 1.5, 2.5, 0
44, 2.5, 1.5, 0
45, 2.5, 2.0, 0
46, 2.5, 2.5, 0
47, 2.5, 0.5, 0
******* E L E M E N T S *************
*ELEMENT, type=T3D2, ELSET=Line1
13, 1, 13
14, 13, 2
*ELEMENT, type=T3D2, ELSET=Line2
15, 2, 14
16, 14, 15
17, 15, 16
18, 16, 12
*ELEMENT, type=T3D2, ELSET=Line3
19, 3, 17
20, 17, 11
*ELEMENT, type=T3D2, ELSET=Line4
21, 11, 18
22, 18, 19
23, 19, 20
24, 20, 10
*ELEMENT, type=T3D2, ELSET=Line5
25, 4, 21
26, 21, 8
*ELEMENT, type=T3D2, ELSET=Line6
27, 8, 22
28, 22, 23
29, 23, 24
30, 24, 9
```

```
*ELEMENT, type=T3D2, ELSET=Line7
31, 5, 25
32, 25, 6
*ELEMENT, type=T3D2, ELSET=Line8
33, 6, 26
34, 26, 27
35, 27, 28
36, 28, 7
*ELEMENT, type=T3D2, ELSET=Line9
37, 1, 29
38, 29, 3
*ELEMENT, type=T3D2, ELSET=Line10
39, 4, 30
40, 30, 5
*ELEMENT, type=T3D2, ELSET=Line11
41, 2, 31
42, 31, 11
*ELEMENT, type=T3D2, ELSET=Line12
43, 11, 32
44, 32, 8
*ELEMENT, type=T3D2, ELSET=Line13
45, 8, 33
46, 33, 6
*ELEMENT, type=T3D2, ELSET=Line14
47, 12, 34
48, 34, 10
*ELEMENT, type=T3D2, ELSET=Line15
49, 10, 35
50, 35, 9
*ELEMENT, type=T3D2, ELSET=Line16
51, 9, 36
52, 36, 7
*ELEMENT, type=CPS4, ELSET=Surface18
53, 1, 29, 37, 13
54, 13, 37, 31, 2
55, 29, 3, 17, 37
56, 37, 17, 11, 31
*ELEMENT, type=CPS4, ELSET=Surface20
57, 2, 14, 38, 31
58, 31, 38, 18, 11
59, 14, 15, 39, 38
60, 38, 39, 19, 18
61, 15, 16, 40, 39
62, 39, 40, 20, 19
63, 16, 12, 34, 40
64, 40, 34, 10, 20
*ELEMENT, type=CPS4, ELSET=Surface22
65, 11, 32, 41, 18
66, 18, 41, 42, 19
67, 19, 42, 43, 20
68, 20, 43, 35, 10
69, 32, 8, 22, 41
70, 41, 22, 23, 42
71, 42, 23, 24, 43
72, 43, 24, 9, 35
```

```
*ELEMENT, type=CPS4, ELSET=Surface24
73, 8, 33, 44, 22
74, 22, 44, 45, 23
75, 23, 45, 46, 24
76, 24, 46, 36, 9
77, 33, 6, 26, 44
78, 44, 26, 27, 45
79, 45, 27, 28, 46
80, 46, 28, 7, 36
*ELEMENT, type=CPS4, ELSET=Surface26
81, 4, 30, 47, 21
82, 21, 47, 33, 8
83, 30, 5, 25, 47
84, 47, 25, 6, 33
```

References

Abgrall, R. (1996). How to prevent pressure oscillations in multicomponent flow calculations: A quasi conservative approach. *Journal of Computational Physics 125*, 150–160. *(p. 97, 103, 115, 116, 134, 296)*

Adams, M., P. Colella, D. T. Graves, J. Johnson, N. Keen, T. J. L. D. F. M. P. McCorquodale, D. M. P. Schwartz, T. Sternberg, and B. V. Straalen (2013). Chombo software package for amr applications - design document. Technical report, Lawrence Berkeley National Laboratory. *(p. 201)*

Allaire, G., S. Clerc, and S. Kokh (2002). A five-equation model for the simulation of interfaces between compressible fluids. *Journal of Computational Physics 181*, 577–616. *(p. 95)*

Almgren, A. S., V. E. Beckner, J. B. Bell, M. S. Day, L. H. Howell, C. C. Joggerst, M. J. Lijewski, A. Nonaka, M. Singer, and M. Zingale (2010). Castro: A new compressible astrophysical solver. i. hydrodynamics and self-gravity. *The Astrophysical Journal 715*(2), 1221. *(p. 201)*

Almgren, A. S., J. B. Bell, M. J. Lijewski, Z. Lukić, and E. V. Andel (2013). Nyx: A massively parallel amr code for computational cosmology. *The Astrophysical Journal 765*(1), 39. *(p. 201)*

Almgren, A. S., J. B. Bell, and W. G. Szymczak (1996). A numerical method for the incompressible navier-stokes equations based on an approximate projection. *SIAM Journal on Scientific Computing 17*, 358–369. *(p. 204)*

Andersson, C., C. Führer, J. Akesson, and M. Gäfvert (2017). Assimulo home page. *(p. 317)*

Antal, S. P., R. T. Lahey, and J. E. Flaherty (1991). Analysis of phase distribution in fully developed laminar bubbly two-phase flow. *International Journal of Multiphase Flow 17*(5), 635–652. *(p. 12)*

Anthonissen, M. J. H., B. A. V. Bennett, and M. D. Smooke (2005). An adaptive multilevel local defect correction technique with application to combustion. *Combustion Theory and Modelling 9*(2), 273–299. *(p. 186)*

Arienti, M., X. Li, M. C. Soteriou, C. A. Eckett, M. Sussman, and R. J. Jensen (2013). Coupled level-set/volume-of-fluid method for simulation of injector atomization. *Journal of Propulsion and Power 29*(1), 147–157. *(p. 18)*

Aris, R. (1962). *Vectors, Tensors and the Basic Equations of Fuid Mechanics*. Dover Publication, Inc. *(p. 78)*

Ashby, S. F. and R. D. Falgout (1996). A parallel multigrid preconditioned conjugate gradient algorithm for groundwater flow simulations. *Nuclear Science and Engineering 124*, 145–159. *(p. 204)*

Audusse, E., F. Bouchut, M.-O. Bristeau, R. Klein, and B. Perthame (2004, July). A fast and stable well-balanced scheme with hydrostatic reconstruction for shallow water flows. *SIAM Journal on Scientific Computing 25*(6), 2050–2065. *(p. 97, 325)*

Augonnet, C., S. Thibault, R. Namyst, and P.-A. Wacrenier (2009, August). StarPU: A Unified Platform for Task Scheduling on Heterogeneous Multicore Architectures. In *Euro-Par 2009*, Delft, Netherlands. *(p. 22)*

Ayguade, E., N. Copty, A. Duran, J. Hoeflinger, Y. Lin, F. Massaioli, X. Teruel, P. Unnikrishnan, and G. Zhang (2009, March). The design of openmp tasks. *IEEE Transactions on Parallel and Distributed Systems 20*(3), 404–418. *(p. 21)*

Bader, M. and C. Zenger (2006). Cache oblivious matrix multiplication using an element ordering based on a peano curve. *Linear Algebra and its Applications 417*(2), 301 – 313. *(p. 210)*

Baer, M. R. and J. W. Nunziato (1986). A two-phase mixture theory for the deflagration-to-detonation transition (ddt) in reactive granular materials. *Int. J. Multiphase Flow 12*(6), 861–889. *(p. 12, 17, 18, 37, 39, 40, 49, 95)*

Balay, S., S. Abhyankar, M. F. Adams, J. Brown, P. Brune, K. Buschelman, L. Dalcin, V. Eijkhout,

W. D. Gropp, D. Kaushik, M. G. Knepley, L. C. McInnes, K. Rupp, B. F. Smith, S. Zampini, H. Zhang, and H. Zhang (2016). PETSc Web page. http://www.mcs.anl.gov/petsc. *(p. 204)*

Bangerth, W., C. Burstedde, T. Heister, and M. Kronbichler (2011, December). Algorithms and data structures for massively parallel generic adaptive finite element codes. *ACM Transactions on Mathematical Software 38*(2), 14/1 – 14/28. *(p. 203, 219)*

Bangerth, W., R. Hartmann, and G. Kanschat (2007). deal.II – a general purpose object oriented finite element library. *ACM Transactions on Mathematical Software 33*(4), 24/1 – 24/27. *(p. 203, 204)*

Bardi, M., R. Payri, L. M. Malbec, G. Bruneaux, L. M. Pichett, J. Manin, T. Bazyn, and C. Genzale (2012). Engine combustion network: comparison of spray development, vaporization, and combustion in different combustion vessels. *Atomization and Sprays 22*(10), 807–842. *(p. 20)*

Baudin, M., C. Berthon, F. Coquel, R. Masson, and Q. H. Tran (2005). A relaxation method for two-phase flow models with hydrodynamic closure law. *Numerische Mathematik 99*(3), 411–440. *(p. 97)*

Baudoin, M., J.-L. Thomas, F. Coulouvrat, and D. Lhuillier (2007). An extended coupled phase theory for the sound propagation in polydisperse concentrated suspensions of rigid particles. *Journal of Acoustical Society of America 121*(6), 3386–3397. *(p. 58)*

Bdzil, J. B., R. Menikoff, S. F. Son, A. K. Kapila, and D. S. Stewart (1999). Two-phase modeling of deflagration-to-detonation transition in granular materials: A critical examination of modeling issues. *Physics of Fluids 11*(2), 378–402. *(p. 39, 41)*

Beau, P.-A. (2006). *Modélisation de l'atomisation d'un jet liquide. Application aux sprays Diesel.* Ph. D. thesis, CORIA - Université de Rouen. *(p. 18)*

Bedford, A. and D. S. Drumheller (1983). Theories of immiscible and structured mixtures. *International Journal of Engineering Science 21*(8), 863. *(p. 42)*

Berdichevsky, V. L. (1983). *Variational Principles of Continuum Mechanics.* Nauka, Moscow. *(p. 42)*

Berdichevsky, V. L. (2009). *Variational Principles of Continuum Mechanics.* Springer. *(p. 41, 42, 43, 45)*

Berger, M. and I. Rigoutsos (1991). An algorithm for point clustering and grid generation. *IEEE Transactions on Systems, Man, and Cybernetics 21*(5), 1278–1286. *(p. 200)*

Berger, M. J. and P. Collela (1989). Local adaptive mesh refinement for shock hydrodynamics. *Journal of Computational Physics 82*(1), 64–84. *(p. 175, 197, 199, 200, 201, 204)*

Berger, M. J. and R. J. Leveque (1998). Adaptive mesh refinement using wave-propagation algorithms for hyperbolic systems. *SIAM Journal on Numerical Analysis 35*(6), 2298–2316. *(p. 201)*

Berger, M. J. and J. Oliger (1984). Adaptive mesh refinement for hyperbolic partial differential equations. *Journal of Computational Physics 53*(3), 484–512. *(p. 175, 182, 186, 199, 200, 204, 299)*

Berlemont, A., T. Ménard, W. Aniszewski, and G. Vaudor (2016). A consistent momentum transport in clsvof method: validation on jet atomization. In *ICMF 2016 - 9th International Conference on Multiphase Flow. (p. 4)*

Bernard-Champmartin, A. and F. De Vuyst (2014). A low diffusive lagrange-remap scheme for the simulation of violent air–water free-surface flows. *Journal of Computational Physics 274*, 19 – 49. *(p. 11, 33, 95, 190, 267, 269, 295)*

Berthon, C. (2005). Stability of the muscl schemes for the euler equations. *Communications in Mathematical Sciences 3*(2), 133–157. *(p. 174, 178)*

Berthon, C. (2006). Why the muscl-hancock scheme is l^1-stable. *Numerische Mathematik 104*, 27–46. *(p. 179, 180)*

Bird, G. A. (1994). Molecular gas dynamics and the direct simulation of gas flows. *Oxford Science Publications 42. (p. 12)*

Boileau, M., C. Chalons, J.-F. Bourgouin, C. Terrier, F. Laurent, S. De Chaisemartin, and M. Massot (2010). Robust numerical schemes for eulerian spray dns and les in two-phase turbulent flows. In *7th International Conference on Multiphase Flows, ICMF 2010*, Number 14.4. 2, pp. 1–15. *(p. 97)*

Boileau, M., C. Chalons, and M. Massot (2015). Robust numerical coupling of pressure and pressureless gas dynamics equations for eulerian spray dns and les. *SIAM Journal on Scientific Computing 37*(1), B79–B102. *(p. 108)*

Bouchut, F. (2004). *Nonlinear Stability of Finite Volume Methods for Hyperbolic Conservation Laws and Well-Balanced Schemes for Sources.* Birkhäuser. *(p. 97, 103, 106, 107, 108, 111, 112, 113, 114, 117, 119, 123, 134, 296, 298)*

Bouchut, F., S. Jin, and X. Li (2003). Numerical approximations of pressureless and isothermal gas dynamics. *SIAM Journal on Numerical Analysis 41*, 135–158. *(p. 225)*

Bourlioux, A. (1995). A coupled level-set volume-of-fluid algorithm for tracking material interfaces. In *Proceedings of the 6th International Symposium on Computational Fluid Dynamics, Lake Tahoe, CA*. *(p. 10)*

Brown, D. L., G. S. Chesshire, W. D. Henshaw, and D. J. Quinlan (1997). Overture: an object oriented software system for solving partial differential equations in serial and parallel environments. In *Proceedings of the eighth SIAM Conference on Parallel Procesing for Scientific Computing*. *(p. 198)*

Brown, P., G. Byrne, and A. Hindmarsh (1989a). VODE: A variable-coefficient ODE solver. *SIAM J. Sci. Stat. Comput. 10*, 1038–1051. *(p. 98)*

Brown, P. N., G. D. Byrne, and A. C. Hindmarsh (1989b). Vode: A variable-coefficient ode solver. *SIAM Journal on Scientific and Statistical Computing 10*(5), 1038–1051. *(p. 149, 172)*

Buffard, T. and S. Clain (2010). Monoslope and multislope muscl methods for unstructured meshes. *Journal of Computational Physics 229*(10), 3745–3776. *(p. 99)*

Burman, E. and L. Sainsaulieu (1995). Numerical analysis of two operator splitting methods for an hyperbolic system of conservation laws with stiff relaxation terms. *Computer Methods in Applied Mechanics and Engineering 128*(3-4), 291–314. *(p. 66)*

Burstedde, C., D. Calhoun, K. Mandli, and A. R. Terrel (2013). Forestclaw: Hybrid forest-of-octrees amr for hyperbolic conservation laws. submitted to International Conference on Parallel Computing - ParCo2013. *(p. 203, 219, 226)*

Burstedde, C. and J. Holke (2016). A tetrahedral space-filling curve for nonconforming adaptive meshes. *SIAM Journal on Scientific Computing 38*(5), 471–503. *(p. 235)*

Burstedde, C., G. Stadler, L. Alisic, L. C. Wilcox, E. Tan, M. Gurnis, and O. Ghattas (2013). Large-scale adaptive mantle convection simulation. *Geophysical Journal International 192*, 889–906. *(p. 204)*

Burstedde, C., L. C. Wilcox, and O. Ghattas (2011). `p4est`: Scalable algorithms for parallel adaptive mesh refinement on forests of octrees. *SIAM Journal on Scientific Computing 33*(3), 1103–1133. *(p. xvi, 24, 203, 204, 207, 209, 210, 211, 212, 214, 215, 217, 297)*

Burstedde, C., L. C. Wilcox, and T. Isaac (2014). *The p4est software for parallel AMR. Howto document and step-by-step examples*. www.p4est.org. *(p. 215)*

Buyevich, Y. A. and I. N. Shchelchkova (1978). Flow of dense suspensions. *Prog. Aerospace Sci. 18*, 121–150. *(p. 58)*

Cahn, J. W. and J. E. Hilliard (1958). Free energy of a nonuniform system. i. interfacial free energy. *Journal of Chemical Physics 28*(2), 258–267. *(p. 10)*

Cahn, J. W. and J. E. Hilliard (1959a). Free energy of a nonuniform system. ii. thermodynamic basis. *Journal of Chemical Physics 30*(5), 1121–1124. *(p. 10)*

Cahn, J. W. and J. E. Hilliard (1959b). Free energy of a nonuniform system. iii. nucleation in a two-component incompressible fluid. *Journal of Chemical Physics 31*(3), 688–699. *(p. 10)*

Calhoun, D. (2014). Adaptive mesh refinement resources. *(p. 201)*

Cargo, P. and A.-Y. le Roux (1994). Un schéma équilibre adapté au modle d'atmosphère avec termes de gravité. *Comptes rendus de l'Académie des Sciences. Série 1, Mathématique 318*(1), 73–76. *(p. 97, 325, 326)*

Caro, F., F. Coquel, D. Jamet, and S. Kokh (2006). A simple finite-volume method for compressible isothermal two-phase flows simulation. *International Journal of Finite Volume 3*(1), www.latp.univ–mrs.fr/IJFV. *(p. 43)*

Chalons, C. and F. Coquel (2005). Navier-stokes equations with several independent pressure laws and explicit predictor-corrector schemes. *Numerische Mathematik 101*(3), 451–478. *(p. 108)*

Chalons, C. and J.-F. Coulombel (2008). Relaxation approximation of the euler equations. *Journal of Mathematical Analysis and Applications 348*(2), 872–893. *(p. 108)*

Chalons, C., M. Girardin, and S. Kokh (2013). Large time step and asymptotic preserving numerical schemes for the gas dynamics equations with source terms. *SIAM Journal on Scientific Computing 35*(6), A2874–A2902. *(p. 96, 97, 325)*

Chalons, C., M. Girardin, and S. Kokh (2016). An all-regime lagrange-projection like scheme for the gas dynamics equations on unstructured meshes. *Communications in Computational Physics 20*(1), 188–233. *(p. 98, 101, 191, 192, 193, 194, 262, 296)*

Chalons, C., P. Kestener, S. Kokh, and M. Stauffert (2016, July). A large time-step and well-balanced

Lagrange-Projection type scheme for the shallow-water equations. working paper or preprint. *(p. 96)*

Chalons, C., M. Massot, and A. Vié (2015). On the eulerian large eddy simulation of dispersed phase flows: an asymptotic preserving scheme for small stokes number flows. *SIAM Multiscale Modeling and Simulation 13*(1), 291–315. submitted on March 2014. *(p. 97)*

Chanteperdrix, G., P. Villedieu, and J. Vila (2002). A compressible model for separated two-phase flows computations. In *ASME Fluid Eng. Div. Summer Meeting 2002*. *(p. 11, 33, 42, 47, 49, 54, 94, 95, 120, 122, 123, 124, 127, 131, 191, 272, 306)*

Chen, G.-Q., C. D. Levermore, and T.-P. Liu (1994). Hyperbolic conservation laws with stiff relaxation terms and entropy. *Communication on Pure and Applied Mathematics 47*(6), 787–830. *(p. 108)*

Chen, J. H. (2011). Petascale direct numerical simulation of turbulent combustion-fundamental insights towards predictive models. *Proceedings of the Combustion Institute 33*, 99–123. *(p. 2)*

Cheng, L. Y., D. A. Drew, and R. T. Lahey (1983). An analysis of wave dispersion, sonic velocity and critical flow in two-phase mixtures. Technical report, U.S. Nuclear Regulatory Commission. *(p. 37, 38, 39, 57, 60, 65)*

Cheng, L. Y., D. A. Drew, and R. T. Lahey (1985, May). An analysis of wave propagation in bubbly two-component, two-phase flow. *Journal of Heat Transfer 107*, 402–408. *(p. 12, 33, 34, 37, 38, 39, 57, 58, 59, 68, 75)*

Chiang, Y.-L., B. van Leer, and K. G. Powell (1992). Simulation of unsteady inviscid flow on an adaptively refined cartesian grid. In *30th Aerospace Sciences Meeting & Exhibit*. *(p. 202)*

Chinnayya, A., A.-Y. L. Roux, and N. Seguin (2004, Avril). A well-balanced scheme for the approximation of the shallow-water equations with topography: the resonance phenomenon. *International Journal of Finite Volume 1*(1), 1–33. *(p. 97, 325)*

CINES (2017). Le supercalculateur occigen. *(p. 226)*

Clain, S., S. Diot, and R. Loubère (2011). A high-order finite volume method for systems of conservation laws-multi-dimensional optimal order detection (mood). *Journal of Computational Physics 230*, 4028–4050. *(p. 235, 298)*

Clerc, S. (2000). numerical simulation of the homogeneous equilibrium model for two-phase flows. *Journal of Computational Physics 161*, 354–375. *(p. 186)*

Cockburn, B. and C.-W. Shu (1989). Tvb runge-kutta local projection discontinuous galerkin finite element method for conservation laws. ii. general framework. *Mathematics of computation 52*(186), 411–435. *(p. 99, 299)*

Cockburn, B. and C.-W. Shu (1998a). The local discontinuous galerkin method for time-dependent convection-diffusion systems. *SIAM Journal on Numerical Analysis 35*(6), 2440–2463. *(p. 99)*

Cockburn, B. and C.-W. Shu (1998b). The runge–kutta discontinuous galerkin method for conservation laws v: multidimensional systems. *Journal of Computational Physics 141*(2), 199–224. *(p. 99)*

Cohen, A., S. M. Kaber, S. Müller, and M. Postel (2003). Fully adaptive multiresolution finite volume schemes for conservation laws. *Mathematics of Computation 72*, 183–225. *(p. 24)*

Colella, P. (1990). Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics 87*(1), 171–200. *(p. 181)*

Commander, K. W. and A. Prosperetti (1989). Linear pressure waves in bubbly liquids: comparison between theory and experiments. *Journal of Acoustical Society of America 85*(2), 732–746. *(p. 68)*

Cordesse, P. (2019). *Study of combustion instabilities in liquid-propellant rocket engines: diffuse interface model and kinetic model coupling for primary atomization simulations*. Ph. D. thesis, Université Paris-Saclay - CentraleSupélec. *(p. 19, 298)*

Curtiss, C. F. and J. O. Hirschfelder (1952, March). Integration of stiff equations. *Proceedings of the US National Academy of Sciences 38*(3), 235–243. *(p. 143, 148)*

Cuzzi, J. N., R. C. Hogan, and K. Shariff (2008). Toward planetesimals: Dense chondrule clumps in the protoplanetary nebula. *The Astrophysical Journal 687*(2), 1432. *(p. 1)*

Dagum, L. and R. Menon (1998, January). Openmp: An industry-standard api for shared-memory programming. *IEEE Comput. Sci. Eng. 5*(1), 46–55. *(p. 21)*

Dahms, R. N. and J. C. Oefelein (2013). On the transition between two-phase and single-phase interface dynamics in multicomponent fluids at supercritical pressures. *Physics of Fluids 25*(9), 092103. *(p. 7)*

Daru, V. and C. Tenaud (2004). High order one-step monotonicity-preserving schemes for unsteady compressible flow calculations. *Journal of Computational Physics 193*, 563–594. *(p. 100, 104, 179)*

Daubechies, I. (1988). Orthonormal bases of compactly supported wavelets. *Comm. Pure and Applied Math. 41*(7), 909–996. *(p. 24)*

Davis, S. F. (1988). Simplified second-order godunov-type methods. *SIAM Journal on Scientific and Statistical Computing 9*(3), 445–473. *(p. 114)*

de Chaisemartin, S. (2009). *Modèles eulériens et simulation numérique de la dispersion turbulente de brouillards qui s'évaporent*. Ph. D. thesis, Ecole Centrale Paris, France. *(p. 13, 225)*

Deiterding, R. (2004). Detonation simulation with the amroc framework. In *Forschung und wissenschaftliches Rechnen: Beiträge zum Heinz-Billing Preis 2003*, pp. 63–77. Göttingen : Ges. für Wiss. Datenverarbeitung, 2004. *(p. 201)*

Deiterding, R., M. O. Domingues, S. M. Gomes, O. Roussel, and K. Schneider (2009). Adaptive multiresolution or adaptive mesh refinement? a case study for 2d euler equations. In *ESAIM: Proceedings*, Volume 29, pp. 28–42. *(p. 25)*

Dellacherie, S. (2005). On a diphasic low-mach number system. *ESAIM: Mathematical Modelling and Numerical Analysis 39*(3), 487–514. *(p. 187)*

Dellacherie, S. (2009). Checkerboard modes and wave equation. In *Proceedings of the Algoritmy 2009 Conference on Scientific Computing*, pp. 71–80. *(p. 193, 265)*

Dellacherie, S. (2010). Analysis of godunov type schemes applied to the compressible euler system at low mach number. *Journal of Computational Physics 229*(4), 978–1016. *(p. 28, 98, 187, 188, 189, 190, 191, 192, 296)*

Dellacherie, S., P. Omnes, and F. Rieper (2010). The influence of cell geometry on the godunov scheme applied to the linear wave equation. *Journal of Computational Physics 229*(14), 5315–5338. *(p. 28, 191)*

Descombes, S., M. Duarte, T. Dumont, T. Guillet, V. Louvet, and M. Massot (2017, April). Task-based adaptive multiresolution for time-space multi-scale reaction-diffusion systems on multi-core architectures. *SMAI Journal of Computational Mathematics 3*, 1–23. *(p. 25)*

Descombes, S., M. Duarte, T. Dumont, F. Laurent, V. Louvet, and M. Massot (2014). Analysis of operator splitting in the non-asymptotic regime for nonlinear reaction-diffusion equations. application to the dynamics of premixed flames. *SIAM Journal on Numerical Analysis 52*(3), 1311–1334. *(p. 96, 97)*

Després, B. and F. Lagoutière (1999, March). Un schéma non linéaire anti-dissipatif pour l'équation d'advection linéaire. *Comptes rendus de l'Académie des Sciences. Série 1, Mathématique 328*, 939–944. *(p. 95)*

Després, B. and F. Lagoutière (2001). Contact discontinuity capturing schemes for linear advection and compressible gas dynamics. *Journal of Scientific Computing 16*, 479–524. *(p. 11, 95, 298)*

Devassy, B. M., C. Habchi, and E. Daniel (2015). Atomization modelling of liquid jets using a two-surface-density approach. *Atomization and Sprays 25*(1), 47–80. *(p. 18, 19, 295)*

Doisneau, F., M. Arienti, and J. C. Oefelein (2017). A semi-lagrangian transport method for kinetic problems with application to dense-to-dilute polydisperse reacting spray flows. *Journal of Computational Physics 329*, 48–72. *(p. 20)*

Doisneau, F., A. Sibra, J. Dupays, A. Murrone, F. Laurent, and M. Massot (2014). Numerical strategy for unsteady two-way coupled polydisperse sprays: application to solid-rocket instabilities. *Journal of Propulsion and Power 30*(3), 727–748. à paraître dans Journal of Propulsion and Power. HAL. *(p. 7)*

Drew, D. A. (1990). Evolution of geometric statistics. *SIAM Journal on Applied Mathematics 50*(3), 649–666. *(p. 12, 77, 78, 81, 82, 83, 295)*

Drew, D. A. and S. L. Passman (1999). *Theory of Multicomponent Fluids*. Applied Mathematical Sciences. Springer. *(p. 12, 34, 35, 36, 37, 57, 58, 66, 68, 81, 82, 83, 85, 295)*

Drui, F., A. Fikl, P. Kestener, S. Kokh, A. Larat, V. L. Chenadec, and M. Massot (2016, March). Experimenting with the p4est library for amr simulations of two-phase flows. In E. Proceedings and Surveys (Eds.), *CEMRACS 2014 – Numerical Modeling of Plasmas*, Volume 53. *(p. 219)*

Drui, F., S. Kokh, A. Larat, M. Massot, and S. Vincent (2017). Comparisons of an incompressible and two compressible methods in the case of the problem of dam breaking. in Preparation. *(p. 278, 293)*

Drui, F., A. Larat, S. Kokh, and M. Massot (2016, July). A hierarchy of simple hyperbolic two-fluid models for bubbly flows. to be submitted to PoF. *(p. 31)*

Duarte, M. (2011, December). *Adaptive numerical methods in time and space for the simulation of*

multi-scale reaction fronts. Theses, Ecole Centrale Paris. https://tel.archives-ouvertes.fr/tel-00667857. *(p. 25)*

Duarte, M. (2012). *Méthodes numériques adaptatives pour la simulation de la dynamique de fronts de réaction multi-échelles en temps et en espace*. Ph. D. thesis, École Centrale Paris. *(p. 96, 99, 252, 299)*

Duarte, M., M. Massot, S. Descombes, C. Tenaud, T. Dumont, V. Louvet, and F. Laurent (2012). New resolution strategy for multiscale reaction waves using time operator splitting, space adaptive multiresolution, and dedicated high order implicit/explicit time integrators. *SIAM J. Sci. Comput. 34*(1), A76–A104. *(p. 25)*

Dubey, A., A. Almgren, J. Bell, M. Berzins, S. Brandt, G. Bryan, P. Colella, D. Graves, M. Lijewski, F. Löffler, et al. (2014). A survey of high level frameworks in block-structured adaptive mesh refinement packages. *Journal of Parallel and Distributed Computing 74*(12), 3217–3227. *(p. 197)*

Dupif, V. (2017). *Modélisation et simulation de l'écoulement diphasique dans les moteurs à propergol solide par une approche eulérienne polydisperses en taille et en vitesse*. Ph. D. thesis, Université Paris-Saclay, préparée à CentraleSupélec. *(p. 99)*

Dupif, V., J. Dupays, A. Larat, and M. Massot (2017). Accurate and robust numerical strategy for axisymetric eulerian modeling of particle laden flows with potential particle trajectory crossing. To be submitted to Journal of Computational Mathematics, SMAI-JCM. *(p. 99)*

Dupif, V., J. Lagarde, M. Boileau, F. Laurent, and M. Massot (2015). Two-size moment eulerian multi-fluid method describing the statistical trajectory crossing: modeling and numerical scheme. in preparation. *(p. 99)*

Embid, P. and M. Baer (1992). Mathematical analysis of a two-phase continuum mixture theory. *Continuum Mechanics and Thermodynamics 4*(4), 279–312. *(p. 41)*

Emre, O., D. Kah, S. Jay, Q.-H. Tran, A. Velghe, S. De Chaisemartin, F. Laurent, and M. Massot (2015). Eulerian Moment Methods for Automotive Sprays. *Atomization and Sprays 25*, 189–254. *(p. 13, 15)*

Ersoy, M., F. Golay, and L. Yushchenko (2013). Adaptive multiscale scheme based on numerical cendity of entropy production for conservation laws. *Central European Journal of Mathematics 11*(8), 1392–1415. *(p. 185)*

Essadki, M. (2017). *Contribution à la modélisation eulérienne unifiée de l'injection : de la zone dense au spray polydispersé*. Ph. D. thesis, Université Paris-Saclay - CentraleSupélec. *(p. 225, 295)*

Essadki, M., S. de Chaisemartin, F. Laurent, and M. Massot (2016). High order moment model for plydisperse evaporating sprays towards interfacial geometry. submitted to SIAM Journal on Applied Mathematics. *(p. 13, 16, 77, 225)*

Essadki, M., S. de Chaisemartin, M. Massot, F. Laurent, A. Larat, and S. Jay (2016, September). Adaptive mesh refinement and high order geometrical moment method for the simulation of polydisperse evaporating sprays. *Oil & Gas Science and Technology 71*(5), 61–86. *(p. 13, 15, 77, 88, 219, 225, 228, 233)*

Essadki, M., F. Drui, S. de Chaisemartin, A. Larat, T. Ménard, and M. Massot (2017, March). Statistical modeling of the gas-liquid interface using geometrical variables: toward a unified description of the disperse and separated phase flows. submitted to IJMF, available on Hal. *(p. 31, 77, 78, 88, 296)*

Essadki, M., J. Jung, A. Larat, M. Pelletier, and V. Perrier (2017, January). A task-driven implementation of a simple numerical solver for hyperbolic conservation laws. to be published in ESAIM: Proceedings and Surveys. *(p. 225, 236, 299)*

Fikl, A. (2014). Adaptive mesh refinement with p4est. Master's thesis, Université Paris XIII. *(p. 175, 219, 224)*

Fleau, S., S. Mimouni, N. Mérigoux, and S. Vincent (2015). Simulations of two-phase flows with a multifield approach. In *Proceedings of CHT-15*. *(p. 20)*

Foldy, L. L. (1945). The multiple scattering of waves. *Physical Review Letters 67*(3 and 4), 107–119. *(p. 57)*

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. *Journal of Computational Physics 227*(6), 3058–3088. *(p. 13)*

Freret, L. and C. P. T. Groth (2015). Anisotropic non-uniform block-based adaptive mesh refinement for three-dimensional inviscid and viscous flows. In *22nd AIAA Computational Fluid Dynamics*

Conference. *(p. 174)*

Freret, L., L. Ivan, H. D. Sterck, and C. P. T. Groth (2017). A high-order finite-volume method with anisotropic amr for ideal mhd flows. In A. S. Forum (Ed.), *55th AIAA Aerospace Sciences Meeting*. *(p. 182, 186, 299)*

Gallice, G. (2002). Solveurs simples positifs et entropiques pour les systèmes hyperboliques avec terme source. *Comptes rendus de l'Académie des Sciences. Série 1, Mathématique 334*, 713–716. *(p. 325)*

Gallice, G. (2003). Positive and entropy stable godunov-type schemes for gas dynamics and mhd equations in lagrangian or eulerian coordinates. *Numerische Mathematik 94*, 673–713. *(p. 325)*

Gallouët, T., P. Helluy, J.-M. Hérard, and J. Nussbaum (2010). Hyperbolic relaxation models for granular flows. *ESAIM: Mathematical Modelling and Numerical Analysis 44*, 371–400. *(p. 41)*

Gavrilyuk, S. (2012). Multiphase flow modeling via hamilton's principle. In F. dell'Isola and S. Gavrilyuk (Eds.), *Variational Models and Methods in Solid and Fluid Mechanics*, Volume 535 of *CISM Courses and Lectures*, pp. 163–210. Springer Vienna. *(p. 12, 45)*

Gavrilyuk, S. and H. Gouin (1999). A new form of governing equations of fluids arising from hamilton's principle. *International Journal of Engineering Science 37*, 1495–1520. *(p. 42, 45)*

Gavrilyuk, S. and R. Saurel (2002). Mathematical and numerical modeling of two-phase compressible flows with micro-inertia. *Journal of Computational Physics 175*, 326–360. *(p. 12, 17, 27, 33, 34, 40, 42, 43, 45, 50, 51, 55, 58, 64, 95, 295, 298)*

Gavrilyuk, S. L. (2005). Acoustic properties of a two-fluid compressible mixture with micro-inertia. *European Journal of Mechanics B/Fluids 24*, 397–406. *(p. 58)*

Gavrilyuk, S. L., H. Gouin, and Y. V. Perepechko (1998). Hyperbolic models of homogeneous two-fluid mixtures. *Meccanica 33*(2), 161–175. *(p. 42, 58)*

Genieys, S. and M. Massot (2001). From navier-stokes equations to oberbeck-boussinesq approximation: a unified approach. Internal report of MAPLY unity (CNRS, UMR 5585). *(p. 186)*

Geurst, J. A. (1986). Variational principles and two-fluid hydrodynamics of bubbly liquid/gas mixtures. *Physica 135*, 455–486. *(p. 42)*

Gingold, R. A. and J. J. Monaghan (1977). Smoothed particle hydrodynamics: theory and application to non-spherical stars. *Monthly Notices of the Royal Astronomical Society 181*, 375–389. *(p. 22)*

Girardin, M. (2014, December). *Asymptotic preserving and all-regime Lagrange-Projection like numerical schemes : application to two-phase flows in low mach regime*. Ph. D. thesis, Université Pierre et Marie Curie. *(p. 97, 98, 134, 173, 191)*

Godlewski, H. and P.-A. Raviart (1996). *Numerical Approximation of Hyperbolic Systems of Conservation Laws*. Springer. *(p. 47, 95, 96, 97, 100, 103, 105, 116, 117, 176)*

Godunov, S. K. (1959). A difference method for numerical calculation of discontinuous solutions of the equations of hydrodynamics. *Matematicheskii Sbornik 89*(3), 271–306. *(p. 97)*

Godunov, S. K. (2008). Reminiscences about numerical schemes. translated version of an original preprint written in russian in 1997, INRIA. *(p. 94, 104)*

Golay, F., M. Ersoy, L. Yushchenko, and D. Sous (2015). Block-based adaptive mesh refinement scheme using numerical density of entropy production for three-dimensional two-fluid flows. *International Journal of Computational Fluid Dynamics 29*(1), 67–81. *(p. xvii, 175, 185, 186, 199, 204, 267, 269, 270, 271, 272)*

Gosse, L. (2000). A well-balanced flux-vector splitting scheme designed for hyperbolic systems of conservation laws with source terms. *Computers and Mathematics with Applications 39*, 135–159. *(p. 97, 325)*

Gosse, L. and A. Y. L. Roux (1996). Un schéma équilibre adapté aux lois de conservation scalaires non-homogènes. *Comptes rendus de l'Académie des Sciences. Série 1, Mathématique 323*, 543–546. *(p. 97, 325)*

Grandgirard, V., Y. Sarazin, P. Angelino, A. Bottino, N. Crouseilles, G. Darmet, G. Dif-Pradalier, X. Garbet, P. Ghendrih, S. Jolliet, G. Latu, E. Sonnendrücker, and L. Villard (2007). Global full-f gyrokinetic simulations of plasma turbulence. *Plasma Physics and Controlled Fusion 49*(12B), B173. *(p. 12)*

Greenberg, J. M. and A. Y. L. Roux (1994, March). A well-balanced scheme for the numerical processing of source terms in hyperbolic equations. *SIAM Journal on Numerical Analysis 33*(1), 1–16. *(p. 97, 325, 326)*

Grenier, N., J.-P. Vila, and P. Villedieu (2013). An accurate low-mach scheme for a compressible two-fluid model applied to free-surface flows. *Journal of Computational Physics 252*, 1 – 19. *(p. 11,*

33, 94, 98, 190, 191, 295, 296)

Gropp, W., E. Lusk, N. Doss, and A. Skjellum (1996). A high-performance, portable implementation of the mpi message passing interface standard. *Parallel Computing 22*(6), 789 – 828. *(p. 21)*

Guillard, H. and C. Viozat (1999). On the behaviour of upwind schemes in the low mach number limit. *Computers and Fluids 28*, 63–86. *(p. 186, 190)*

Gunney, B. T. N. and R. W. Anderson (2016). Advances in patch-based adaptive mesh refinement scalability. *Journal of Parallel and Distributed Computing 89*, 65–84. *(p. 198, 200, 201)*

Haider, F. (2009). *Discrétisation en maillage non structuré général et applications LES*. Ph. D. thesis. 2009PA066171. *(p. 100)*

Hairer, E. (2017). Fortran and matlab codes. *(p. 317)*

Hairer, E., S. P. Nørsett, and G. Wanner (1987). *Solving Ordinary Differential Equations I*. Berlin: Springer-Verlag. Nonstiff Problems. *(p. 98)*

Hairer, E. and G. Wanner (1996). *Solving Ordinary Differential Equations II* (2nd ed.). Berlin: Springer-Verlag. Stiff and Differential-Algebraic Problems. *(p. 98, 135, 172, 317)*

Harten, A. (1983). High resolution schemes for hyperbolic conservation laws. *Journal of Computational Physics 49*(3), 357 – 393. *(p. 100)*

Harten, A. (1994). Adaptive multiresolution schemes for shock computations. *Journal of Computational Physics 115*, 319–338. *(p. 24, 182, 186, 200, 237, 252, 299)*

Harten, A. (1995). Multiresolution algorithms for the numerical solution of hyperbolic conservation laws. *Comm. Pure and Applied Math. 48*, 1305–1342. *(p. 24)*

Hayatdavoodi, M. and R. C. Ertekin (2016). Review of wave loads on coastal bridge decks. *Applied Mechanics Reviews 68*(3), 030802–030802–16. *(p. 1, 293)*

Helluy, P., F. Golay, J.-P. Caltagirone, P. Lubin, S. Vincent, D. Drevard, R. Marcer, P. Fraunié, N. Seguin, S. Grilli, A.-C. Lesage, A. Dervieux, and O. Allain (2005). Numerical simulations of wave breaking. *ESAIM: Mathematical Modelling and Numerical Analysis 39*(3), 591–607. *(p. 272, 275)*

Henshaw, W. D. and D. W. Schwendeman (2003). An adaptive numerical scheme for high-speed reactive flow on overlapping grids. *Journal of Computational Physics 191*, 420–447. *(p. 198)*

Herrmann, M. (2008). A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics 227*, 2674–2706. *(p. 18)*

Herrmann, M. (2010a). Detailed numerical simulations of the primary atomization of a turbulent liquid jet in crossflow. *Journal of Engineering for Gas Turbines and Power 132*, 061506–1–061506–10. *(p. 18)*

Herrmann, M. (2010b). A parallel eulerian interface tracking/lagrangian point particle multi-scale coupling procedure. *Journal of Computational Physics 229*, 745–759. *(p. 18)*

Heuveline, V. and R. Rannacher (2003). Duality-based adaptivity in the hp-finite element method. *Journal of Numerical Mathematics 11*(2), 95–113. *(p. 200)*

Hindmarsh, A. (1983). ODEPACK, a systematized collection of ODE solvers. In *Scientific computing (Montreal, Que., 1982)*, pp. 55–64. New Brunswick, NJ: IMACS. *(p. 98)*

Hirsch, C. (1990). *Numerical Computation of internal and external flows, Vol II: computational methods for inviscid and viscous flows*. Wiley. *(p. 104, 190)*

Hirt, C., A. Amsden, and J. Cook (1974). An arbitrary lagrangian-eulerian computing method for all flow speeds. *Journal of Computational Physics 14*(3), 227 – 253. *(p. 23)*

Hirt, C. W. and B. D. Nichols (1981). Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of Computational Physics 39*, 201–225. *(p. 9)*

Hodge, W. V. D. (1989). *The Theory and Applications of Harmonic Integrals*. Cambridge University Press. *(p. 189)*

Holmes, D. G. and S. D. Connell (1989). Solution of the 2d navier-stokes equations on unstructured adaptive grids. In *9th Computational Fluid Dynamics Conference*. *(p. 23)*

Hopkins, P. F. (2015). A new class of accurate, mesh-free hydrodynamic simulation methods. *Monthly Notices of the Royal Astronomical Society 450*(1), 53. *(p. 23)*

Hornung, R. D. and S. R. Kohn (2002). Managing application complexity in the samrai object-oriented framework. *Concurrency and Computation 14*(5), 347–368. *(p. 201)*

Howell, L. H. and J. B. Bell (1997). An adaptive mesh projection method for viscous incompressible flow. *SIAM Journal on Scientific Computing 18*(4), 996–1013. *(p. 204)*

Huang, W., Y. Ren, and R. D. Russell (1994). Moving mesh methods based on moving mesh partial

differential equations. *Journal of Computational Physics 113*(2), 279 – 290. *(p. 23)*

Hülsemann, F., B. Bergen, and U. Rüde (2003). Hierarchical hybrid grids as basis for parallel numerical solution of pde. In *Euro-Par 2003 Parallel Processing*, pp. 840–843. *(p. xv, 199)*

Hundsdorfer, W. and J. G. Verwer (2003). *Numerical Solution of Time-Dependent Advection-Diffusion-Reaction Equations*, Volume 33 of *Springer Series in Computational Mathematics*. Springer. *(p. 97)*

Isaac, T., C. Burstedde, and O. Ghattas (2012, May). Low-cost parallel algorithms for 2:1 octree balance. In *Parallel Distributed Processing Symposium (IPDPS), 2012 IEEE 26th International*, pp. 426–437. *(p. 214, 217)*

Isaac, T., C. Burstedde, L. C. Wilcox, and O. Ghattas (2015). Recursive algorithms for distributed forests of octrees. *SIAM Journal on Scientific Computing 37*(5), C497–C531. *(p. xvi, 205, 207, 208, 215, 217)*

Ishii, M. (1975). *Thermo-Fluid dynamics theory of two-phase flow*. Eyrolles. *(p. 8, 12, 34, 58)*

Jacqmin, D. (1999). Calculation of two-phase navier-stokes flows using phase-field modeling. *Journal of Computational Physics 155*, 96–127. *(p. 10)*

Jamet, D. (2010). Diffuse interface models in fluid mechanics. *GdR CNRS documentation*. *(p. 10)*

Jamet, D., O. Lebaigue, N. Courtis, and J. M. Delhaye (2001). The second gradient method for the direct numerical simulation of liquid-vapor flows with phase change. *Journal of Computational Physics 169*, 624–651. *(p. 10)*

Jánosi, I. M., D. Jan, K. G. Szabó, and T. Tél (2004). Turbulent drag reduction in dam-break flows. *Experiments in Fluids 37*, 219–229. *(p. 276, 282)*

Jemison, M., M. Sussman, and M. Arienti (2014). Compressible, multiphase semi-implicit method with moment of fluid interface representation. *Journal of Computational Physics 279*, 182–217. *(p. 10)*

Ji, H., F.-S. Lien, and E. Yee (2010, November). A new adaptive mesh refinement data structure with an application to detonation. *Journal of Computational Physics 229*(23), 8981–8993. *(p. 202)*

Jiang, G. and C. W. Shu (1996). Efficient implementation of weighted eno schemes. *Journal of Computational Physics 126*, 202–228. *(p. 99)*

Jin, S. (1999). Efficient asymptotic-preserving (ap) schemes for some multiscale kinetic equations. *SIAM Journal on Scientific Computing 21*(2), 441–454. *(p. 97)*

Jin, S. and Z. Xin (1995). The relaxation schemes for systems of conservation laws in arbitrary space dimensions. *Communication on Pure Applied Mathematics 48*(3), 235–276. *(p. 94, 103, 108, 109)*

Kah, D. (2010). *Prise en compte des aspects polydispersés pour la modélisation d'un jet de carburant dans les moteurs à combustion interne*. Ph. D. thesis, Ecole Centrale Paris. *(p. 13, 295)*

Kah, D., F. Laurent, M. Massot, and S. Jay (2012). A high order moment method simulating evaporation and advection of a polydisperse spray. *Journal of Computational Physics 231*(2), 394–422. *(p. 15, 225)*

Kamimoto, T. and M. hoan Bae (1988, 02). High combustion temperature for the reduction of particulate in diesel engines. In *SAE Technical Paper*. SAE International. *(p. xiii, 3)*

Kapila, A. K., R. Menikoff, J. B. Bdzil, S. F. Son, and D. S. Stewart (2001). Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations. *Physics of Fluids 13*(10), 3002–3024. *(p. 17, 41)*

Kermani, M. J., A. G. Gerber, and J. M. Stockie (2003). Thermodynamically based moisture prediction using roe's scheme. In *The 4th Conference of Iranian Aerospace Society*. *(p. 99)*

Khokhlov, A. M. (1998, July). Fully threaded tree for adaptive refinement fluid dynamics simulations. *Journal of Computational Physics 143*(2), 519–543. *(p. 202, 203, 204)*

Klainerman, S. and A. Majda (1981). Singular limits of quasilinear hyperbolic systems with large parameters and the incompressible limit of compressible fluids. *Communication on Pure Applied Mathematics 34*, 481–524. *(p. 186)*

Kokh, S. and F. Lagoutière (2010). An anti-diffusive numerical scheme for the simulation of interfaces between compressible fluids by means of a five-equation model. *Journal of Computational Physics 229*(8), 2773–2809. *(p. 11, 95, 103, 116, 117, 123, 134, 269, 298)*

Koltakov, S. and O. B. Fringer (2013). Moving grid method for numerical simulation of stratified flows. *International Journal for Numerical Methods in Fluids 71*(12), 1524–1545. *(p. 23)*

Koshizuka, S., H. Tamako, and Y. Oka (1995). A particle method for incompressible viscous flow with fluid fragmentations. *Computational Fluid Dynamics Journal 4*(1), 29–46. *(p. 267)*

Landau, L. D. and E. M. Lifshitz (1966). *Fluid Mechanics* (3rd ed.), Volume 6 of *Course of Theoretical Physics*. Pergamon Press. *(p. 59)*

Lasheras, J. C. and E. J. Hopfinger (2000). Liquid jet instability and atomization in a coaxial gas stream. *Annual Review of Fluid Mechanics 32*, 275–308. *(p. xiii, 6)*

Laurent, F. and M. Massot (2001). Multi-fluid modelling of laminar polydisperse spray flames: origin, assumptions and comparison of sectional and sampling methods. *Combustion Theory and Modelling 5*, 537–572. *(p. 13, 19)*

Laurent, F., A. Sibra, and F. Doisneau (2016). Two-size moment eulerian multi-fluid model: a flexible and realizable high-fidelity description of polydisperse moderately dense evaporating sprays. *Communications in Computational Physics 20*(4), 902–943. *(p. 13)*

Lax, P. D. and B. Wendroff (1960). Systems of conservation laws. *Communication on Pure and Applied Mathematics 13*, 217–237. *(p. 104)*

Le Chenadec, V. (2012). *A Stable and Conservative Framework for Detailed Numerical Simulation of Primary Atomization*. Ph. D. thesis, Stanford University. *(p. 10)*

Le Touze, C. (2015, December). *Couplage entre modèles diphasiques à "phases séparées" et à "phase dispersée" pour la simulation de l'atomisation primaire en combustion cryotechnique*. Ph. D. thesis, Université Nice Sophia Antipolis. *(p. 19, 99)*

Le Touze, C., A. Murrone, and H. Guillard (2015). Multislope MUSCL method for general unstructured meshes. *Journal of Computational Physics 284*, 389–418. *(p. 19)*

Lebas, R. (2007). *Modélisation Eulérienne de l'Atomisation Haute Pression. Influences sur la Vaporisation et la Combustion Induite*. Ph. D. thesis, INSA de Rouen - Université de Rouen. *(p. 18)*

Lee, C. S. and S. W. Park (2002). An experimental and numerical study on fuel atomization characteristics of high-pressure diesel injection sprays. *Fuel 81*, 2417–2423. *(p. 7)*

Lemoine, A. (2014). *Décomposition de Hodge-Helmoltz discrète*. Ph. D. thesis, Université de Bordeaux. *(p. 189)*

Leroy, V., A. Strybulevych, J. H. Page, and M. G. Scanlon (2008, April). Sound velocity and attenuation in bubbly gels measured by transmission experiments. *Journal of Acoustical Society of America 123*(4), 1931–1940. *(p. 67, 68, 69, 70)*

Leveque, R. J. (1992). *Numerical Methods for Conservation Laws*. Birkhäuser Basel. *(p. 106)*

Leveque, R. J. (2004). *Finite-Volume Methods for Hyperbolic Problems*. Cambridge University Press. *(p. 96, 97, 100, 103, 117, 175, 176, 177, 179, 194, 201, 203, 219, 302)*

Lhuillier, D., C.-H. Chang, and T. G. Theofanous (2013). On the quest for a hyperbolic effective-field model of disperse flows. *Journal of Fluid Mechanics 731*, 184–194. *(p. 37)*

Lhuillier, D., T. G. Theofanous, and M.-S. Liou (2010). *Handbook of Nuclear Engineering*, Volume 1. Springer. *(p. 64)*

Li, X.-s. and C.-w. Gu (2008). An all-speed roe-type scheme and its asymptotic analysis of low mach number behaviour. *Journal of Computational Physics 227*, 5144–5159. *(p. 190)*

Liboff, R. (1969). *Introduction to the theory of kinetic equations*. Wiley. *(p. 13)*

Lions, P.-L. and N. Masmoudi (1998). Incompressible limit for a viscous compressible fluid. *Journal de Mathématiques Pures et Appliquées 77*, 585–627. *(p. 186)*

Liou, M.-S. (1996). A sequel to ausm:ausm+. *Journal of Computational Physics 129*, 364–382. *(p. 190)*

Liu, T.-P. (1987). Hyperbolic conservation laws with relaxation. *Communications in Mathematical Physics 108*, 153–175. *(p. 108, 109)*

Lobovsky, L., E. Botia-Vera, F. Castellana, J. Mas-Soler, and A. Souto-Iglesias (2014, July). Experimental investigation of dynamic pressure loads during dam break. *Journal of Fluids and Structures 48*, 407–434. *(p. xvii, 275, 276, 277, 278, 280, 282, 283, 284, 286)*

Lucy, L. B. (1977, December). A numerical approach to the testing of the fission hypothesis. *The Astronomical Journal 82*(12), 1013–1024. *(p. 22)*

MacNeice, P., K. M. Olson, C. Mobarry, R. de Fainchtein, and C. Packer (2000). Paramesh: A parallel adaptive mesh refinement community toolkit. *Computer Physics Communications 126*, 330–354. *(p. 200, 203)*

Mandumpala Devassy, B. (2013). *Atomization Modeling of Liquid Jets using an Eulerian-Eulerian Model and a Surface Density Approach*. Ph. D. thesis, Université d'Aix-Marseille. *(p. 18, 20)*

Marchuk, G. I. (1968). Some application of splitting-up methods to the solution of mathematical physics problems. *Applications of Mathematics 13*(2), 103–132. *(p. 96)*

Marmottant, P. and E. Villermaux (2004). On spray formation. *Journal of Fluid Mechanics 498*, 73–111. *(p. 4, 19)*

Martelot, S. L. (2013). *Contribution à la simulation numérique directe de l'ébullition*. Ph. D. thesis,

Université Aix-Marseille. *(p. 190)*

Martin, J. C. and W. J. Moyce (1952, March). An experimental study of the collapse of liquid columns on a rigid horizontal plane. *Philosophical Transactions of the Royal Society of London. Series A, MAthematical and Physical Sciences 244*(882), 312–324. *(p. xvii, 275, 276, 277, 279, 285, 286)*

Massot, M., F. Laurent, D. Kah, and S. de Chaisemartin (2010a). A robust moment method for evaluation of the disappearance rate of evaporating sprays. *SIAM J. Appl. Math. 70*, 3203–3234. *(p. 13, 15)*

Massot, M., F. Laurent, D. Kah, and S. de Chaisemartin (2010b). A robust moment method for evaluation of the disappearance rate of evaporating sprays. *SIAM Journal on Applied Mathematics 70*, 3203–3234. *(p. 225)*

McGraw, R. (1997). Description of aerosol dynamics by the quadrature method of moments. *Aerosol Science and Technology 27*, 255–265. *(p. 13)*

Mead, L. R. and N. Papanicolaou (1984). Maximum entropy in the problem of moments. *Journal of Mathematical Physics 25*, 2404–2417. *(p. 16)*

Ménard, T., S. Tanguy, and A. Berlemont (2007, May). 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*(2), 510–524. *(p. 10)*

Minion, M. L. (1996). A projection method for locally refined grids. *Journal of Computational Physics 127*, 158–178. *(p. 204)*

Morel, C. (2007). On the surface equations in two-phase flows and reacting single-phase flows. *International Journal of Multiphase Flow 33*, 1045–1073. *(p. 18, 77, 78, 81)*

Morel, C. (2015). *Mathematical Modeling of Disperse Two-Phase Flows*, Volume 114 of *Fluid Mechanics and Its Applications*. Springer. *(p. 77, 81, 295)*

Morton, G. M. (1966). A computer oriented geodetic data base; and a new technique in file sequencing. Technical report, IBM Ltd. *(p. 209)*

Moureau, V. and G. Lartigue (2015). Yales2 www. coria-cfd. fr. *(p. 197)*

Müller, S. (2003). *Adaptive Multiscale Schemes for Conservation Laws*, Volume 27. Springer. Ed. T. J. Barth, M. Griebel, D. E. Keyes, R. M. Nieminen , D. Roose and T. Schlick. *(p. 24, 252)*

Murrone, A. and H. Guillard (2008). Behavior of upwind scheme in the low mach number limit: Iii. preconditioned dissipation for a five equation two-phase model. *Computers and Fluids 37*(10), 1209–1224. *(p. 99)*

Nguyen, T. T., F. Laurent, R. O. Fox, and M. Massot (2016). Solution of population balance equations in applications with fine particles: mathematical modeling and numerical schemes. *J. Comp. Phys. 325*, 129–156. *(p. 13, 15)*

Nonaka, A., A. S. Almgren, J. B. Bell, M. J. Lijewski, C. M. Malone, and M. Zingale (2010). Maestro: An adaptive low mach number hydrodynamics algorithm for stellar flows. *The Astrophysical Journal Supplement Series 188*(2), 358. *(p. 201)*

Osher, S. (1988). Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics 79*(1), 12–49. level set. *(p. 9)*

Paillere, H., C. Viozat, A. Kumbaro, and I. Toumi (2000). Comparison of low mach number models for natural convection problems. *Heat and Mass Transfer 36*, 567–573. *(p. 186)*

Pauchon, C. and P. Smereka (1992). Momentum interactions in dispersed flow: an averaging and variational approach. *International Journal of Multiphase Flow 18*(1), 65. *(p. 42)*

Pelanti, M., F. Bouchut, and A. Mangeney (2011). A riemann solver for single-phase and two-phase shallow flow models based on relaxation. relations with roe and {VFRoe} solvers. *Journal of Computational Physics 230*(3), 515 – 550. *(p. 97)*

Pianet, G., S. Vincent, J. Leboi, J. P. Caltagirone, and M. Anderhuber (2010). Simulating compressible gas bubbles with a smooth volume tracking 1-fluid method. *International Journal of Multiphase Flow 36*, 273–283. *(p. 190, 275, 278, 279, 285, 286)*

Plesset, M. S. and A. Prosperetti (1977, January). Bubble Dynamics and Cavitation. *Annual Review of Fluid Mechanics 9*(1), 145–185. *(p. 17, 58, 59)*

Pope, S. B. (1988). The evolution of surfaces in turbulence. *International Journal of Engineering Science 26*(5), 445–469. *(p. 35, 77, 78, 80, 81, 82, 83, 84)*

Popinet, S. (2003, September). Gerris: a tree-based adaptive solver for the incompressible euler equations in complex geometries. *Journal of Computational Physics 190*(2), 572–600. *(p. 202, 203, 204)*

Popinet, S. (2015). A quadtree-adaptive multigrid solver for the serre-green-naghdi equations. *Journal of Computational Physics 302*, 336–358. *(p. 202, 203)*

Prosperetti, A. (1977, January). Thermal effects and damping mechanisms in the forced radial oscillations of gas bubbles in liquids. *Journal of Acoustical Society of America 61*(1), 17–27. *(p. xiii, 52, 57, 58, 59, 60, 61, 62, 64, 65, 68)*

Prothero, A. and A. Robinson (1974). On the stability and accuracy of one-step methods for solving stiff systems of ordinary differential equations. *Mathematics of Computation 28*(125), 145–162. *(p. 142)*

Rannacher, R. and B. Vexler (2005). A priori estimates for the finite element discretization of elliptic parameters identification problems with pointwise measurements. *SIAM Journal on Control and Optimization 44*, 1844–1863. *(p. 200)*

Rayleigh, L. (1917). On the pressure developed in a liquid during the collapse of a spherical cavity. *Philosophical Magazine 6*(34), 94–98. *(p. 17, 58, 59)*

Reitz, R. D. and F. V. Bracco (1982). Mechanism of atomization of a liquid jet. *Physics of Fluids 25*(10), 1730–1742. *(p. 4)*

Richardson, L. F. (1911). The approximate arithmetical solution by finite differences of physical problems including differential equations, with an application to the stresses in a masonry dam. *Philosophical Transactions of the Royal Society of London. Series A, Mathematical and Physical Sciences 210*(459-470). *(p. 200)*

Ritter, A. (1892). Die fortpflanzung der wasserwellen. *Zeitschrift des Vereines Deutscher Ingenieure 36*, 947–954. *(p. 276)*

Rossinelli, D., B. Hejazialhosseini, D. G. Spampinato, and P. Koumoutsakos (2011). Multicore/multi-GPU accelerated simulations of multiphase compressible flows using wavelet adapted grids. *SIAM J. Sci. Comput. 33*(2), 512–540. *(p. 25)*

Rozar, F. (2015, November). *Towards highly scalable parallel simulations for turbulent plasma physics*. Theses, Université de Bordeaux. *(p. 22)*

Sampath, R. S., S. S. Adavani, H. Sundar, I. Lashuk, and G. Biros (2008, Nov). Dendro: Parallel algorithms for multigrid and amr methods on 2:1 balanced octrees. In *2008 SC - International Conference for High Performance Computing, Networking, Storage and Analysis*, pp. 1–12. *(p. 204)*

Saurel, R. and R. Abgrall (1999). A multiphase godunov method for compressible multifluid and multiphase flows. *Journal of Computational Physics 150*, 425–467. *(p. 18, 95, 103)*

Scardovelli, R. and S. Zaleski (1999). Direct numerical simulation of free-surface and interfacial flow. *Annual Review in Fluid Mechanics 31*, 567–603. *(p. 2)*

Schneider, K. and O. Vasilyev (2010). Wavelet methods in computational fluid dynamics. *Annual Review of Fluid Mechanics 42*, 473–503. *(p. 24)*

Schochet, S. (1986). The compressible euler equations in a bounded domain: existence of solutions and the incompressible limit. *Communications in Mathematical Physics 104*, 49–75. *(p. 186)*

Schochet, S. (1994). Fast singular limits of hyperbolic pdes. *Journal of Differential Equations 114*, 476–512. *(p. 186, 189)*

Schwartz, P., M. Barad, P. Colella, and T. Ligocki (2006). A cartesian grid embedded boundary method for the heat equation and poisson's equation in three dimensions. *Journal of Computational Physics 211*(2), 531–550. *(p. 198)*

Secchi, P. (2000). On the singular incompressible limit of inviscid compressible fluids. *Journal of Mathematical Fluid Mechanics 2*, 107–125. *(p. 186)*

Serre, D. (2007, january). Systems of conservation laws with dissipation. Lecture notes. *(p. 47, 138)*

Shu, C.-W. (2013). A brief survey on discontinuous galerkin methods in computational fluid dynamics. *Advances in Mechanics 43*(6), 541–553. *(p. 235)*

Sibra, A. (2015, November). *Eulerian Multi-Fluid modeling and simulation of evaporation and combustion of polydisperse sprays in solid rocket motors*. Theses, Université Paris-Saclay. *(p. xiii, 23)*

Silberman, E. (1957). Sound velocity and attenuation in bubbly mixtures measured in standing wave tubes. *Journal of Acoustical Society of America 29*, 925–933. *(p. 58, 67, 68, 69)*

Spitzer, Jr, L. (1943, July). Acoustic properties of gas bubbles in a liquid. Technical report, Columbia University, Division of War Research. *(p. 57)*

Stansby, P. K., A. Chegini, and T. C. D. Barnes (1998). The initial stages of dam-break flow. *Journal of Fluid Mechanics 374*, 407–424. *(p. 275, 277, 278, 280, 286, 288, 291)*

Stewart, H. B. and B. Wendroff (1984). Two-phase flow: models and methods. *Journal of Computational Physics 56*, 363–409. *(p. 37)*

Stoufflet, B., J. Periaux, F. Fezoui, and A. Dervieux (1987). Numerical simulation of 3-d hypersonic euler flows around space vehicles using adapted finite elements. In *25th AIAA Aerospace Sciences Meeting*. *(p. 99)*

Strang, G. (1968). On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis 5*, 506–517. *(p. 96)*

Stuhmiller, J. H. (1977). The influence of interfacial pressure on the character of two-phase flow model equations. *International Journal of Multiphase Flow 3*, 551–560. *(p. 37)*

Suliciu, I. (1990). On modelling phase transitions by means of rate-type constitutive equations, schock wave structure. *International Journal of Engineering Science 1*, 829–841. *(p. 103, 108, 117, 296)*

Sundar, H., R. S. Sampath, S. S. Adavani, C. Davatzikos, and G. Biros (2007). Low-constant parallel algorithms for finite element simulations using linear octrees. In ACM (Ed.), *Proceedings of the ACM/IEEE International Conference for High Performance Computing, Networking, Storage and Analysis*. *(p. 202, 207)*

Sundar, H., R. S. Sampath, and G. B. Biros (2008). Bottom up construction and 2:1 balance refinement of linear octrees in parallel. *SIAM Journal on Scientific Computing 30*, 2675–2708. *(p. 202, 207, 214)*

Sussman, M. and A. G. Puckett (2000). A coupled level set and volume-of-fluid method for computing 3d axisymmetric incompressible two-phase flows. *Journal of Computational Physics 162*(2), 301–337. *(p. 10)*

Sweby, P. K. (1984). High resolution schemes using flux limiters for hyperbolic conservation laws. *SIAM Journal of Numerical Analysis 21*(5), 995–1011. *(p. 178)*

Téman, R. (1969a). Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires. i. *Archive for Rational Mechanics and Analysis 32*, 135–153. *(p. 96)*

Téman, R. (1969b). Sur l'approximation de la solution des équations de navier-stokes par la méthode des pas fractionnaires. ii. *Archive for Rational Mechanics and Analysis 33*, 377–385. *(p. 96)*

Temkin, S. (2005). *Suspension Acoustics - An introduction to the physics of suspensions*. Cambridge University Press. *(p. 51, 64)*

Teshukov, V. M. (2000, September). Kinetic model of bubbly flow. *Journal of Applied Mechanics and Technical Physics 41*(5), 879–886. *(p. 17)*

Teshukov, V. M. and S. L. Gavrilyuk (2002). Kinetic model for the motion of compressible bubbles in a perfect fluid. *European Journal of Mechanics B/Fluids 21*, 469–491. *(p. 17)*

Teyssier, R. (2002). Cosmological hydrodynamics with adaptive mesh refinement. a new high resolution code called ramses. *Astronomy and Astrophysics 385*, 337–364. *(p. 198, 202, 203, 225)*

Tiwari, A., J. B. Freund, and C. Pantano (2013). A diffuse interface model with immiscibility preservation. *Journal of Computational Physics 252*, 290–309. *(p. 20)*

TOP500 (2017, April). Top500 the list. *(p. xiii, 2)*

Toro, E. F. (2009). *Riemann Solvers and Numerical Methods for Fluid Dynamics - A Practical Introduction* (3rd ed.). Springer. *(p. 97, 103, 107, 112, 114, 123, 124, 179, 255, 296, 309)*

Trotter, H. (1959). On the product of semi-groups of operators. *Proc. Am. Math. Soc. 10*, 545–551. *(p. 96)*

Truesdell, C. (1969). *Rational Thermodynamics*. McGraw-Hill. *(p. 39, 40)*

Trulio, J. G. and K. R. Trigger (1961). Numerical solution of the one-dimensional lagrangian hydrodynamic equations. Technical report, California. Univ., Livermore, CA (United States). Lawrence Radiation Lab. *(p. 23)*

Tryggvason, G., B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y.-J. Jan (2001). A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics 169*, 708–759. *(p. 9)*

Tsoutsanis, P., I. W. Kokkinakis, L. Könözsy, D. Drikakis, R. J. R. Williams, and D. L. Youngs (2015). Comparison of structured- and unstructured-grid, compressible and incompressible methods using the vortex pairing poblem. *Computer Methods in Applied Mechanics and Engineering 293*, 207–231. *(p. 191)*

Tsuji, Y., T. Kawaguchi, and T. Tanaka (1993). Discrete particle simulation of two-dimensional fluidized bed. *Powder Technology 77*(1), 79 – 87. *(p. 1)*

Turkel, E. (1987). Preconditioned methods for solving the incompressible and low speed compressible

equations. *Journal of Computational Physics 72*, 277–298. *(p. 190)*

Turkel, E. (1993). Review of preconditioning methods for fluid dynamics. *Applied Numerical Mathematics 12*, 257–284. *(p. 190)*

van Leer, B. (1979). A second-order sequel to godunov's method. *Journal of Computational Physics 32*, 101–136. *(p. 94, 178)*

van Leer, B. (1984). On the relation between the upwind-differencing schemes of godunov, engquist-osher and roe. *SIAM Journal on Scientific and Statistical Computing 5*(1), 1–20. *(p. 178, 179)*

van Wijngaarden, L. (1972). One dimensional flow of liquids containing small gas bubbles. *Annual Review of Fluid Mechanics 4*, 369–396. *(p. 68)*

Vazquez-Gonzalez, T. (2016, April). *Schémas numériques mimétiques et conservatifs pour la simulation d'écoulements multiphasiques compressibles*. Ph. D. thesis, Université Paris-Saclay - CentraleSupélec. *(p. 23, 37)*

Venditti, D. A. and D. L. Darmofal (2000). Adjoint error estimation and grid adaptation for functional outputs: Application to quasi-one-dimensional flow. *Journal of Computational Physics 164*, 204–227. *(p. 200)*

Vié, A., F. Laurent, and M. Massot (2013). Size-velocity correlations in high order moment methods for polydisperse evaporating sprays: modeling and numerical issues. *J. Comp. Phys. 237*, 277–310. *(p. 15)*

Villermaux, E. (2004). Unifying ideas on mixing and atomization. *New Journal of Physics 6*(125), 1–19. *(p. 19)*

Vincent, S., J.-P. Caltagirone, P. Lubin, and T. N. Randrianarivelo (2004). An adaptive augmented lagrangian method for three-dimensional multimaterial flows. *Computers and Fluids 33*, 1273–1289. *(p. 278)*

Vincent, S., J. Larocque, J.-P. Caltagirone, and G. Pianet (2007). Adaptive augmented lagrangian techniques for simulating unsteady multiphase flows. In *Sixth International Conference Multiphase Flow, Leipzig, Germany*, pp. 203. *(p. 278)*

Vincent, S., A. Sarthou, J.-P. Caltagirone, F. Sonilhac, P. Février, C. Mignot, and G. Pianet (2011). Augmented lagrangian and penalty methods for the simulation of two-phase flows interacting with moving solids. application to hydroplanning flows interacting with real tire tread patterns. *Journal of Computational Physics 230*, 956–983. *(p. 278)*

von Neumann, J. and R. D. Richtmeyer (1950). A method for the numerical calculation of hydrodynamic shocks. *Journal of Applied Physics 21*, 232–237. *(p. 22)*

Weisstein, E. W. (2017). Gauss-bonnet formula. *(p. 86)*

Whitham, G. B. (1974). *Linear and Nonlinear Waves*. Wiley-Interscience. *(p. 49, 66, 109)*

Williams, F. A. (1958). Spray combustion and atomization. *Physics of Fluids 1*(6), 541–545. *(p. 14, 77)*

Wissink, A. M., D. Hysom, and R. D. Hornung (2003). Enhancing scalability of parallel structured amr calculations. In *Proceedings of the 17th ACM International Conference on Supercomputing*, pp. 336–347. *(p. 200)*

Wood, A. B. (1930). *A textbook of Sound*. The Macmillan Company. *(p. 47, 58)*

Woodward, P. and P. Colella (1984). The numerical simulation of two-dimensional fluid flow with strong shocks. *Journal of Computational Physics 54*, 115–173. *(p. 190)*

Yuan, C., F. Laurent, and R. Fox (2012). An extended quadrature method of moments for population balance equations. *Atomization and Sprays 51*(1), 1–23. *(p. 13, 15)*

Zamansky, R., F. Coletti, M. Massot, and A. Mani (2016). Turbulent thermal convection driven by heated inertial particles. *Journal of Fluid Mechanics 809*, 390–4437. *(p. 12)*

université PARIS-SACLAY | École doctorale de mathématiques Hadamard (EDMH)

Titre : Modélisation et simulation Eulériennes des écoulements diphasiques à phases séparées et dispersées : développement d'une modélisation unifiée et de méthodes numériques adaptées au calcul massivement parallèle

Mots clefs : Modèle diphasique, solveurs de Riemann approchés, adaptation dynamique de maillages, principe variationnel, écoulements bas-Mach, rupture de barrage.

Résumé : Dans un contexte industriel, l'utilisation de modèles diphasiques d'ordre réduit est nécessaire pour pouvoir effectuer des simulations numériques prédictives d'injection de combustible liquide dans les chambres de combustion automobiles et aéronautiques, afin de concevoir des équipements plus performants et moins polluants. Le processus d'atomisation du combustible, depuis sa sortie de l'injecteur sous un régime de phases séparées, jusqu'au brouillard de gouttelettes dispersées, est l'un des facteurs clés d'une combustion de bonne qualité. Aujourd'hui cependant, la prise en compte de toutes les échelles physiques impliquées dans ce processus nécessite une avancée majeure en termes de modélisation, de méthodes numériques et de calcul haute performance (HPC). Ces trois aspects sont abordés dans cette thèse. Premièrement, des modèles de mélange, dérivés par le principe variationnel de Hamilton et le second principe de la thermodynamique sont étudiés. Ils sont alors enrichis afin de pouvoir décrire des pulsations des interfaces au niveau de la sous-échelle. Des comparaisons avec des données expérimentales dans un contexte de milieux à bulles permettent de vérifier la cohérence physique des modèles et de valider la méthodologie. Deuxièmement, une stratégie de discrétisation est développée, basée sur une séparation d'opérateur, permettant la résolution indépendante de la partie convective des systèmes à l'aide de solveurs de Riemann approchés standards et les termes sources à l'aide d'intégrateurs d'équations différentielles ordinaires. Ces différentes méthodes répondent aux particularités des systèmes diphasiques compressibles, ainsi qu'au choix de l'utilisation de maillages adaptatifs (AMR). Pour ces derniers, une stratégie spécifique est développée : il s'agit du choix de critères de raffinement et de la projection de la solution d'une grille à une autre (plus fine ou plus grossière). Enfin, l'utilisation de l'AMR dans un cadre HPC est rendue possible grâce à la bibliothèque AMR p4est, laquelle a montré une excellente scalabilité jusqu'à plusieurs milliers de coeurs de calcul. Un code applicatif, CanoP, a été développé et permet de simuler des écoulements fluides avec des méthodes de volumes finis sur des maillages AMR. CanoP pourra être utilisé pour des futures simulations d'atomisation liquide.

Title: Eulerian modeling and simulations of separated and disperse two-phase flows: development of a unified modeling approach and associated numerical methods for highly parallel computations.

Keywords: Two-phase model, approximate Riemann solvers, adaptive mesh refinement, variational principle, low-Mach flows, dam-break.

Abstract: In an industrial context, reduced-order two-phase models are used in predictive simulations of the liquid fuel injection in combustion chambers and help designing more efficient and less polluting devices. The combustion quality strongly depends on the atomization process, starting from the separated phase flow at the exit of the nozzle down to the cloud of fuel droplets characterized by a disperse-phase flow. Today, simulating all the physical scales involved in this process requires a major breakthrough in terms of modeling, numerical methods and high performance computing (HPC). These three aspects are addressed in this thesis. First, we are interested in mixture models, derived through Hamilton's variational principle and the second principle of thermodynamics. We enrich these models, so that they can describe sub-scale pulsations mechanisms. Comparisons with experimental data in a context of bubbly flows enables to assess the models and the methodology. Based on a geometrical study of the interface evolution, new tracks are then proposed for further enriching the mixture models using the same methodology. Second, we propose a numerical strategy based on finite volume methods composed of an operator splitting strategy, approximate Riemann solvers for the resolution of the convective part and specific ODE solvers for the source terms. These methods have been adapted so as to handle several difficulties related to two-phase flows, like the large acoustic impedance ratio, the stiffness of the source terms and low-Mach issues. Moreover, a cell-based Adaptive Mesh Refinement (AMR) strategy is considered. This involves to develop refinement criteria, the setting of the solution values on the new grids and to adapt the standard methods for regular structured grids to non-conforming grids. Finally, the scalability of this AMR tool relies on the p4est AMR library, that shows excellent scalability on several thousands cores. A code named CanoP has been developed and enables to solve fluid dynamics equations on AMR grids. We show that CanoP can be used for future simulations of the liquid atomization.

Université Paris-Saclay
Espace Technologique / Immeuble Discovery
Route de l'Orme aux Merisiers RD 128 / 91190 Saint-Aubin, France