On the numerical simulation of evaporating two-phase flows using sharp interface capturing methods

Victor Boniou

université
PARIS-SACLAY

On the numerical simulation of evaporating two-phase flows using sharp interface capturing methods

Simulation numérique d'écoulements diphasiques évaporants par des méthodes de capture d'interface raides

Thèse de doctorat de l'université Paris-Saclay

École doctorale n° 579, Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMaG)
Spécialité de doctorat : Mécanique des fluides
Graduate School : Sciences de l'ingénierie et des systèmes.
Référent : CentraleSupélec

Thèse préparée dans l'unité de recherche du Laboratoire EM2C (Université Paris-Saclay, CNRS, CentraleSupélec)
sous la direction de Sébastien DUCRUIX, directeur de recherche
et le co-encadrement de Aymeric VIÉ, maître de conférence
et Thomas SCHMITT, chargé de recherche.

Thèse soutenue à Gif-Sur-Yvette, le 9 décembre 2021, par

Victor BONIOU

Composition du jury

Christian TENAUD Directeur de recherche, CNRS, EM2C	Président
Olivier DESJARDINS Professeur agrégé, Université Cornell, SSMAE	Rapporteur & Examinateur
Stéphane POPINET Directeur de recherche, CNRS, UPMC	Rapporteur & Examinateur
Sébastien TANGUY Maître de conférences, Université de Toulouse, IMFT	Examinateur
Vincent MOUREAU Chargé de recherche, CNRS, CORIA	Examinateur
Sébastien DUCRUIX Directeur de recherche, CNRS, EM2C	Directeur de thèse
Thomas SCHMITT Chargé de recherche, CNRS, EM2C	Coencadrant
Aymeric VIÉ Maître de conférences, CentraleSupélec, CNRS, EM2C	Coencadrant

THESE DE DOCTORAT
NNT : 2021UPAST128

Remerciements

Tout d'abord, je souhaite remercier mon jury de thèse, une vraie équipe de "superstars" dont je me suis grandement inspiré et qui ont accepté d'évaluer mes travaux. J'en suis sincèrement honoré.
Merci à Christian Tenaud d'avoir présider mon jury, j'espère avoir encore l'occasion d'interagir avec toi sur des questionnements de numérique et pourquoi pas du compressible ?!
Merci à Stéphane Popinet et Olivier Desjardins pour avoir pris le temps de lire en détail mon manuscrit et pour vos précieux retours de rapporteurs. J'ai beaucoup appris à travers vos travaux et j'ai été particulièrement touché par vos commentaires bienveillants sur le travail que j'ai conduit pendant ces trois années.
Merci également à Vincent Moureau et Sébastien Tanguy, pour les discussions et interactions que l'ont a pu avoir à plusieurs étapes de ma thèse sur tous les aspects subtiles et sournois des méthodes numériques et pour l'examen de mon travail finalisé.

Je finis ce jury par mon équipe de direction, à commencer par Sébastien Ducruix qui a accepté de diriger cette thèse purement numérique et qui a su me faire confiance pendant ces trois années.
Merci à Thomas Schmitt, pour ta passion pour la recherche et ta curiosité sans limite qui m'ont plusieurs fois poussé dans mes retranchements au cours de longues discussion "sur la vie" comme tu le dis si bien. J'ai adoré travailler avec toi, autant pour tes qualités scientifiques qu'humaines.
Enfin merci à Aymeric Vié, le deuxième pilote de ce tandem d'encadrants exceptionnels, qui a toujours pris le temps de répondre à mes interrogations, suivre mes avancées, à toute heure du jour ou de la nuit, la semaine comme le week-end (on se rappelle des dernières relectures de manuscrit le samedi après-midi !). Si je souhaite poursuivre dans la recherche c'est en grande partie grâce à toi.

Maintenant, parlons du laboratoire EM2C.
Merci à Jean-Michel pour sa gentillesse et sa patience lorsqu'il s'agit de compiler un code ou répondre à des questions tordues sur les normes de Fortran.
En parlant de code, je souhaite également remercier Milan Pelletier, mon partenaire de développement pendant une bonne année sans qui TITAN V1.0 n'aurait pas pu voir le jour et être aussi structuré.
Merci également "aux filles du secrétariat", Nathalie, Noï et Brigitte, qui ont été mes mères adoptives sans qui je n'aurais sûrement pas réussi à me réinscrire ne serait-ce qu'en

deuxième année de doctorat. Vous faites un sacré travail, et je voulais insister sur votre implication sans faille dans le bon fonctionnement de ce laboratoire.

Ensuite, il est difficile de parler du laboratoire sans parler de ses doctorants.
Je suis arrivé dans un bureau plein de vieux, qui étaient tous en fin de thèse avec beaucoup de sagesse (ou pas ?). Merci à Amanda, Lorella, Giampaolo, Abi, Stefano, Jan et bien évidemment Léo qui ont su m'accueillir et me faire me sentir chez moi. Ensuite, le bureau a subi un grand nombre de bouleversements avec vos départs successifs et ce fût à mon tour d'accueillir les nouveaux, Artur, Maxime, Marie, Subash et maintenant Julie. J'ai passé de très bons moments avec vous (je vous promets que je m'améliorerais en origami).
Je n'oublie pas mes partenaires de galères qui ont commencé en même temps que moi, je vous lâche en premier, désolé, mais j'ai hâte de vous voir réussir : David, Victorien, Roxanne, Luc, Preethi, Corentin.
Je vais également parler de la team piscine : Suzanne, Junghwa, Constantin, Guilhem et Arnaud avec qui j'ai pu nager les midis d'hiver comme de printemps avec l'espoir d'avoir le "summer body", sans succès.
Je n'oublie pas bien sûr Ulysse, avec qui j'ai partagé de nombreux trajets de navette, j'ai toujours pris un grand plaisir à discuter avec toi de musique, de voyage ou autres (quand je ne regardais pas de drama coréen avec Junghwa ...).
J'en ai forcément oublié, et je m'en excuse par avance.

Enfin je finirais par remercier mes proches. Merci à mes amis pour les week-ends et vacances que l'on a pu passer loin de la thèse et très (trop ?) proche de la connerie. Vous m'avez permis de trouver l'équilibre parfait pour arriver jusqu'au bout ! Merci enfin à ma famille, mon frère ainsi que mes parents avec qui j'ai pu revivre les années lycée au cours des confinements passés dans la maison familiale.

Abstract

High fidelity simulation of evaporating two-phase flows is a growing subject of interest with the emergence of multiple solvers and various associated numerical methods.
This is a challenging subject because of the strong discontinuities of flow quantities at the interface that require attention in numerical methods design.
For sharp Eulerian methods, several strategies have shown their capability to reproduce the physics without phase change using Volume-of-fluid (VOF) or Level-set (LS).
With phase change, the interface jumps must account for the mass and heat transfers, which requires far more attention numerical methods design. Few contributions can be found, and dedicated studies are necessary to reach the same level of understanding as for non-evaporating flows.
The objective of this work is to compare possible solutions for simulating such flows with sharp Eulerian methods. To this end, they are all implemented in the same cartesian-grid solver and share the same unified Low-Mach framework to allow fair comparisons.

First, the numerical representation of an interface is investigated for four popular interface capturing methods with up-to-date numerics: the Volume-of-fluid (VOF), the standard Level-Set (SLS), the accurate conservative Level-Set (ACLS), and the coupled Level-Set and Volume-of-fluid (CLSVOF). The comparison is made on canonical cases with imposed velocity fields.
In this configuration, even if all methods show consistent results with mesh refinement, CLSVOF appears to be the most accurate method, and its versatility could be a strong asset for more complex configurations.

Then, all methods are embedded into the unified low-Mach framework without phase change. This framework includes a sharp transport of the interface, a well-balanced surface tension discretization, and a consistent mass-momentum transport, which allows capillary-driven simulations with high-density ratios.
Specific attention is given to surface-tension-driven cases. Then, complex test cases such as droplet collision or shear-induced breakup are investigated.
All methods give comparable results for simple test cases if the numerical resolution is sufficiently high. For the complex configurations, especially for the shear-induced breakup, all methods give very different results, which call for new investigation for their ability to capture the onset of instabilities.

Finally, the unified framework is extended to phase change using a two-scalar approach for energy and species equations for VOF and SLS methods. The associated challenges are investigated through numerical experiments to justify the numerical choices made in the unified framework.
Specific attention is devoted to analyzing of numerical errors in the reconstruction of interface quantities, showing the importance of extrapolation methods and evaluating quantities at the interface location.
A quantitative analysis is given on canonical evaporation problems for which analytical solutions are available. Both VOF and SLS methods show mesh convergence, but with opposite trends for the static evaporation case: while SLS overestimates the interface regression, VOF underestimates it.

The last test case is a convected droplet in a quiescent flow, which implies convection, interface deformation, and non-homogeneous vaporization. In this case, the importance of the consistency between momentum and continuity is highlighted by considering different literature techniques for the continuity equation. It is shown that non-consistent formulations lead to an increase in the droplet velocity. The results for SLS suggest that the shared framework is too constraining, and a dedicated framework must be used. Finally, as VOF equations can be rigorously derived from the conservation equations, there is no ambiguity in the discrete continuity equation. We then consider the results to be representative of the physics.

Résumé

La simulation des écoulements diphasiques évaporant est un sujet en plein essor avec l'émergence de multiples solveurs et méthodes numériques associés.
C'est un sujet difficile en raison des discontinuités de l'écoulement à l'interface qui nécessitent des méthodes numériques adaptées.
Parmi les méthodes Eulériennes « sharp », plusieurs stratégies ont montré leur capacité à reproduire la physique sans changement de phase, que ce soit en Volume-of-Fluid (VOF) ou en Level-Set (LS).
Avec le changement de phase, les sauts d'interface doivent tenir compte des transferts de masse et de chaleur. Peu de contributions existent, et des études spécifiques sont nécessaires pour atteindre le même niveau de compréhension que pour les écoulements sans évaporation.
L'objectif de ce travail est de comparer les solutions possibles pour simuler de tels écoulements. Toutes les méthodes sont implémentées dans le même solveur et partagent le même cadre unifié Low-Mach pour permettre des comparaisons équitables.

La simulation d'une interface est d'abord étudiée pour quatre méthodes : le VOF, la standard LS (SLS), l'accurate conservative LS (ACLS), et le coupled LS-VOF (CLSVOF). La comparaison est effectuée sur des cas avec champ de vitesse imposé.
Dans ce cadre, même si toutes les méthodes présentent des résultats cohérents et convergents en maillage, le CLSVOF semble être la méthode la plus précise, et sa polyvalence pourrait être un atout important pour des cas plus complexes.

Ensuite, les méthodes sont intégrées dans le même cadre unifié, sans changement de phase. Ce cadre comprend un transport précis de l'interface, une discrétisation équilibrée de la tension de surface et un transport masse-quantité de mouvement cohérent qui permet des simulations avec un rapport de densité élevé et des effets capillaires.
Une attention particulière est accordée aux cas avec tension superficielle. Puis, des cas tests complexes de collision de gouttes ou de rupture induite par cisaillement sont étudiés.
Les méthodes donnent des résultats comparables pour les cas simples, si la résolution numérique est suffisamment élevée. Pour les cas complexes, en particulier pour la rupture induite par cisaillement, les méthodes donnent des résultats différents, ce qui nécessite des études sur la capacité des méthodes à saisir le début des instabilités.

Enfin, le cadre unifié est étendu au changement de phase en utilisant une approche à deux

scalaires pour les équations d'énergie et d'espèces, pour les méthodes VOF et SLS. Les nouveaux défis associés sont étudiés numériquement afin de justifier les choix effectués.
Une attention particulière est consacrée aux erreurs numériques dans la reconstruction des quantités d'interface, montrant l'importance des méthodes d'extrapolation et de l'évaluation des quantités à la position de l'interface.
Une analyse quantitative est donnée sur les problèmes d'évaporation canoniques avec solutions analytiques. Les méthodes VOF et SLS montrent une convergence en maillage, mais avec des tendances opposées pour le cas d'évaporation statique : alors que la SLS surestime la régression de l'interface, le VOF la sous-estime.

Le dernier cas test est une goutte convectée dans un écoulement au repos, qui implique une convection, une déformation de l'interface et une vaporisation non-homogène. L'importance de la cohérence entre l'équation de quantité de mouvement et de continuité est mise en évidence en considérant différentes techniques de la littérature pour l'équation de continuité. Il est montré que les formulations incohérentes conduisent à une augmentation de la vitesse de la goutte. Les résultats pour la SLS suggèrent que le cadre unifié est trop contraignant, et qu'un cadre dédié doit être utilisé. Enfin, comme les équations VOF peuvent être rigoureusement dérivées des équations de conservation, il n'y a aucune ambiguïté dans l'équation de continuité discrète, et nous considérons que les résultats sont représentatifs de la physique.

Contents

Introduction

Spray combustion in engines

Evaporation is a natural phenomenon ubiquitous in industrial processes such as food drying [43], spray cooling [99], or spray combustion [201]. This last application is a subject of growing interest for transportation applications as liquid fuel results in a crucial gain in energy density for storage purposes.
In an engine, the fuel is injected as a liquid jet in the combustion chamber and needs to turn into its gas phase to initiate combustion, as shown in Fig. 1. To this end, it is first atomized to increase the surface exchange between the liquid and the gas by breaking up the jet into a spray of droplets of various sizes. These droplets are then transported into the flow, interacting with turbulence, and finally evaporate to feed the combustion process. The study of spray combustion thus implies a large variety of physical interactions and a large range of scales, making both experiments and simulations very challenging.

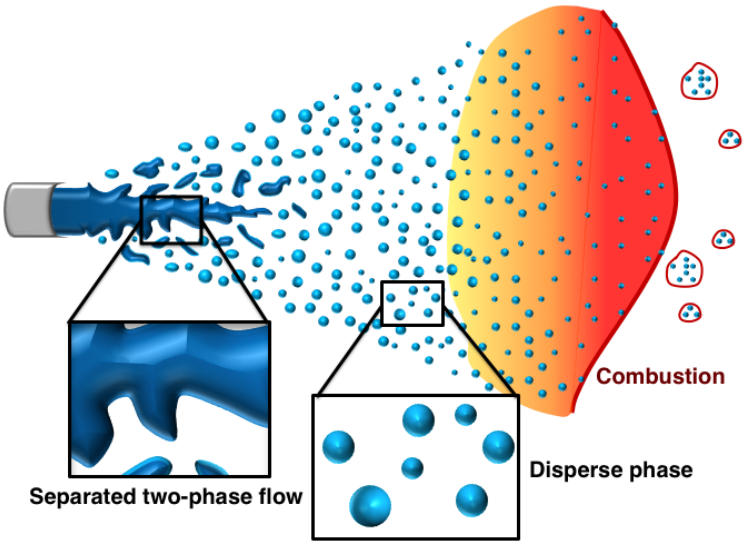


Figure 1: *Schematic representation of the different phases of the spray evolution (from* [130]*).*

Toward accurate droplet vaporization models

This work takes part in a research project which aims to improve the numerical simulations of spray combustion. One aspect is the study of the coupling between the evaporation of droplets with turbulence and combustion.
Because of the large size ratio between a combustor and a droplet, realistic simulations of combustors are performed using large eddy simulations (LES). They model droplet-flow interactions in the sub-cell by using correlations from experiments or one-dimensional derivations under strong assumptions on the flow and the topology of the droplet. Recent contributions have highlighted the sensitivity of such models to the different correlations used to determine the heat and mass exchanges. For instance, Shashank et al. [96] showed that the evaporation is strongly dependent on the gas composition used to evaluate the heat capacity in the film region (see Fig. 2)

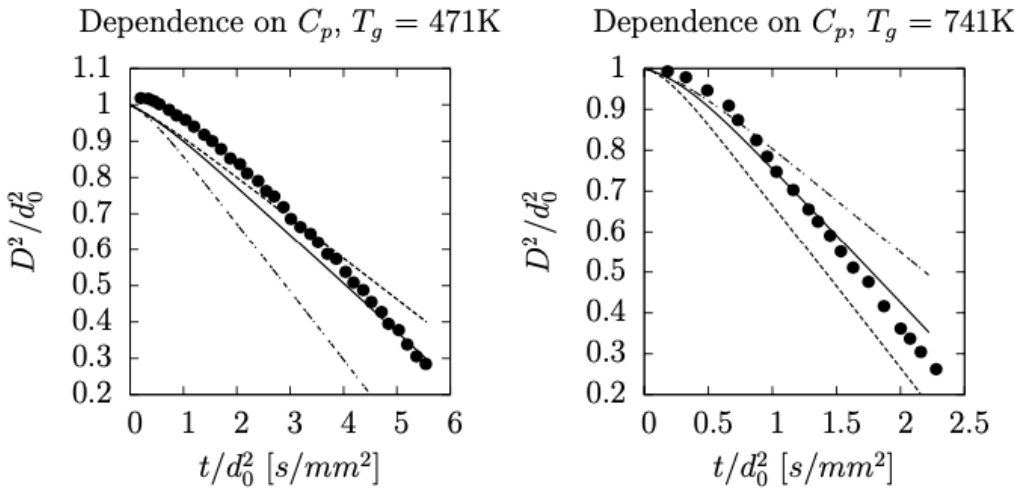


Figure 2: *Sensitivity of heptane drop diameter to the gas composition used to evaluate C_p (from [96]). Experiments (dots) are compared with model solutions for different evaluation of C_p (lines).*

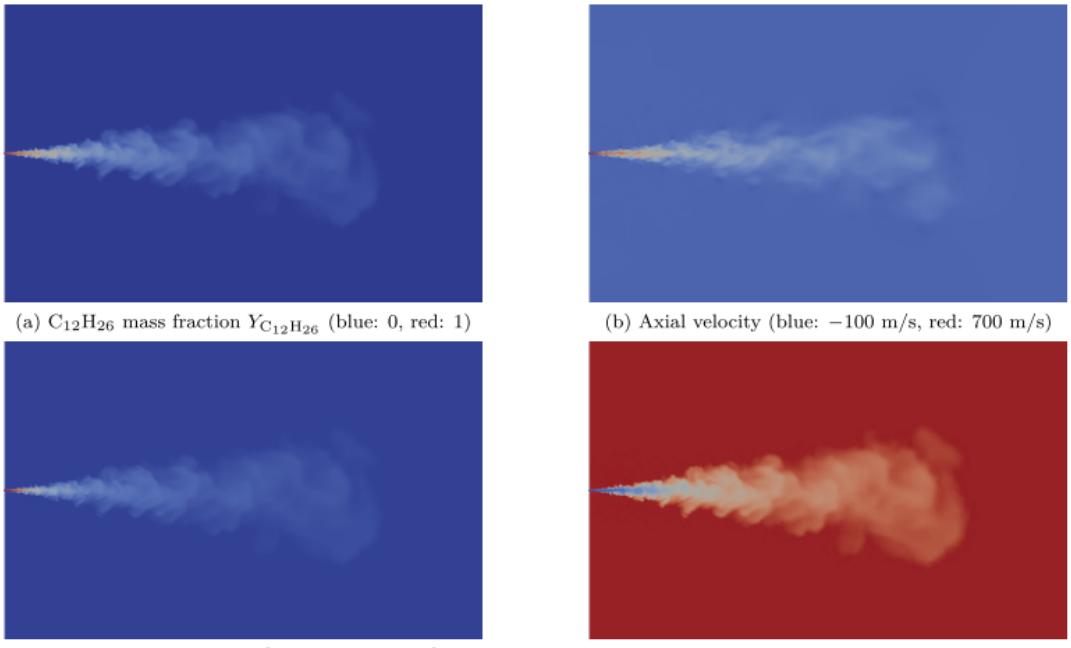
A strategy to improve these models is to acquire experimental data as in [132; 32]. However, the studies are performed on simplified configurations where droplets are supported by a fiber which is far from the conditions of a combustion chamber.

High-fidelity simulation of two-phase flows

To improve the vaporization models, another solution is to use high-fidelity simulations of droplet vaporization. The growing capacities of massively parallel computations give access to more and more possibilities in the numerical resolution of such problems.
In the literature, different methods can be employed to solve two-phase flows, which can be classified into two main interface representations

1. The diffuse interface methods (DIM) define the interface as a region where quantity variations are stiff but continuous. Then, they do not need to represent the interface

location explicitly. They are often used to study compressible flows as they rely on enriched thermodynamic treatments in the interfacial region where classical thermodynamics is not valid.
A first class of DIM encompasses the multifluid methods [191] which describes the interface as an artificially diffused region. They are based on an ensemble averaging of the properties of the phases. The resulting sets of governing equations are hyperbolic, provided that convex thermodynamic closures are used.
On the other hand, the phase-field methods [26] are based on an advanced thermodynamic description of the interface. Physical properties of the interface (thickness, surface tension) are naturally embedded within the governing equations and thermodynamics. However, the resulting system of equations is generally complex to solve numerically, especially for compressible flows.
An illustration of the simulation of a Diesel injection using a multifluid method is given in Fig. 3 from the work of [164].



(a) $C_{12}H_{26}$ mass fraction $Y_{C_{12}H_{26}}$ (blue: 0, red: 1)

(b) Axial velocity (blue: -100 m/s, red: 700 m/s)

(c) Density (blue: 21 kg/m^3, red: 650 kg/m^3, log scale)

(d) Temperature (blue: 360 K, red: 920 K)

Figure 3: *Simulation of Diesel injection in a high pressure chamber with AVBP solver using a 3-equation multifluid model* [164].

2. The sharp interface method (SIM) uses an infinitely thin interface which is a moving surface with a given position. With this representation, the fluid properties are discontinuous at the interface and the interface needs to be located accurately to apply the correct jump conditions at the interface.
   Front tracking methods [223] explicitly transport Lagrangian markers all belonging

to the interface. This allows to be highly accurate in the interface topology, such as interface normal and curvature but it requires special treatments in non-trivial cases of coalescence or atomization [172; 222].
The other type of SIM is the interface capturing method relying on an Eulerian representation of the interface through the transport of a color function. Volume-of-fluid (VOF) methods [83] are based on the resolution of the volume fraction of one phase in each numerical cell. This method has the advantage of being intrinsically mass-conserving. However, the interface is not explicitly known and must be retrieved through a reconstruction strategy, such as the PLIC method [181]. Moreover, topology information requires dedicated attention, as differentiating the sharp volume fraction field can lead to huge errors. Instead of solving for the volume fraction, level set (LS) [153] methods transport a function that is related implicitly to the interface. This function is chosen to be smoother than the volume fraction, thus being easier to solve numerically and giving access to better topology information. However, attention must be drawn to conservation properties.
An application on the simulation of a liquid jet in a crossflow using accurate conservative LS is provided in Fig. 4 from the work of [110].

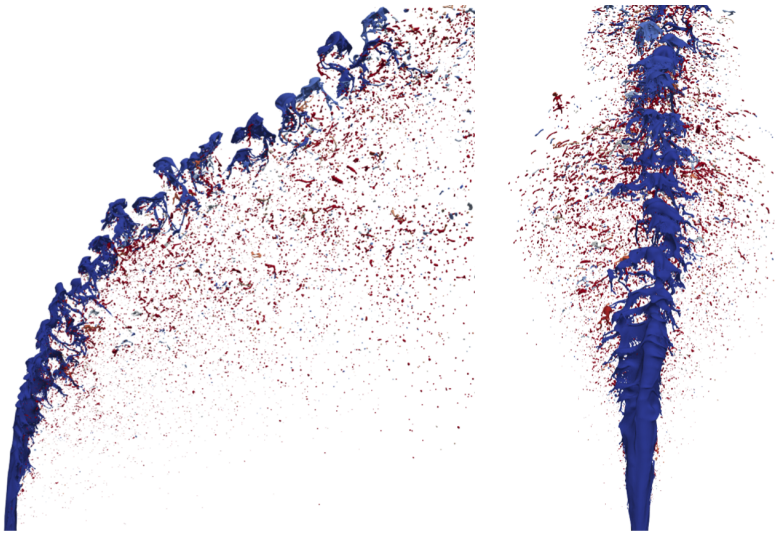


Figure 4: *Simulation of a liquid jet in a crossflow simulated with the YALES2 solver using an ACLS method* [110]

As, the objective of this work is the study of low Mach evaporating two-phase flows, compressible formulations are not needed. Moreover the DIM generally requires a large number of points to properly resolve an interface and thus a droplet Moreover, the hypothesis of infinitely thin interfaces is a reasonable assumption in the thermodynamic conditions of the target applications. Thus, SIMs are preferred in this work.

Accounting for phase change in two-phase flow simulations

When evaporation comes into place, the numerical methods based on SIM have to be adapted to treat additional discontinuities, which constitutes challenging issues. In the late '90s, the pioneer works of Juric et al. [95] with an FT method, Welch et al. [227] using VOF or Son et al. [204] with LS have paved the way for simulations with phase change. These works only contained the boiling phenomenon as the set of equations and the phase-change modelling was limited to mono-component mixtures.
Some years later, Tanguy et al. [218] proposed a LS solver for evaporation simulations with a bi-component gas phase. The method was able to perform simulations of convected droplets in 2D axisymmetric configurations. One year later, a solver based on VOF proposed by Schlottke et al. [195] performed 3D simulations of evaporation of convected droplets subject to high deformations.
Since these contributions, the domain of two-phase flow simulations has been very active with a noticeable gain in interest for the simulation of evaporation [198; 183; 85; 29; 159; 192; 179]. This increasing research activity resulted in considerable improvements in the robustness and accuracy of the methods in the last decade, which allowed to simulate more complex configurations such as evaporation in turbulence [4; 52; 126], combustion [86] or supersonic flows [101].
High-fidelity simulations of droplet evaporation represent a promising approach to improve the understanding of the interaction between evaporation and turbulence as illustrated in Fig. 5. In Fig. 5a a methodology based on LS proposed by [4] shows the capability to simulate the evaporation of a group of droplets subject to a turbulent flow. In Fig. 5b, the coupled LS-VOF methodology of [126] performs the simulation of a liquid evaporating in Homogeneous Isotropic Turbulence (HIT).

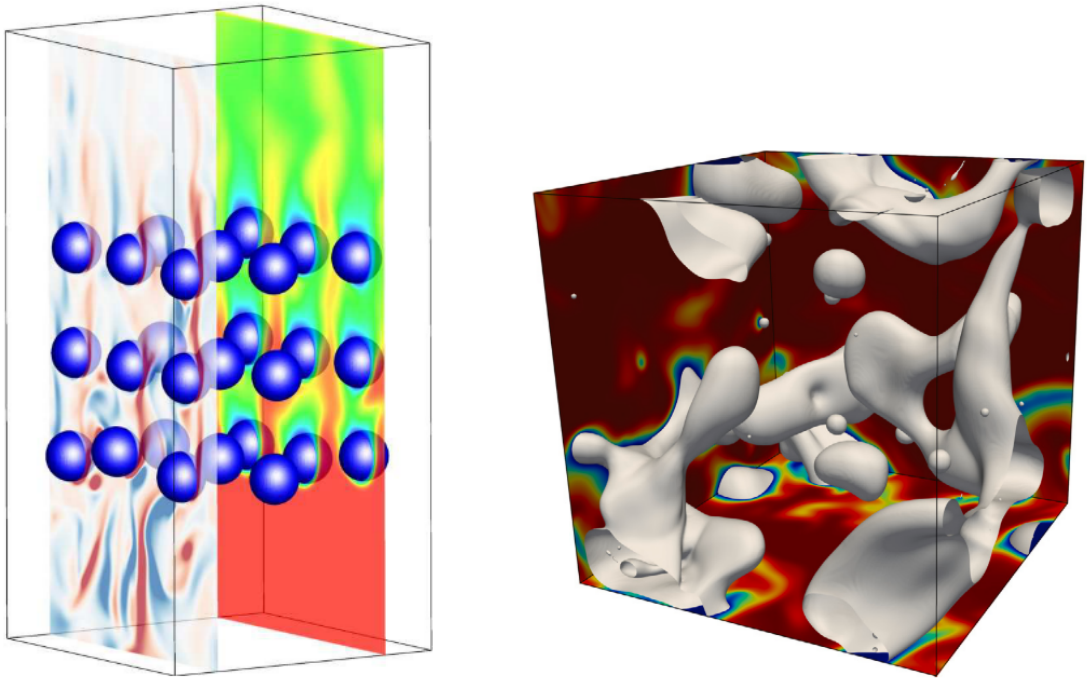
The literature presented here shows a very large spectrum of numerical methods, indicating that no consensus exist on the best methodology to achieve evaporation simulations. Based on this observation, it is then not clear to choose for

- The sharp interface method between FT, VOF, and LS as they all seem to be well-suited for evaporation simulations.
- The methodology to properly couple the interface transport with the mass and momentum equations.
- The integration of the phase-change modelling in the solver.

This is the main motivation for developing our own numerical framework in which it is possible to implement and compare different approaches for the simulation of two-phase flows with phase change.

A dedicated solver

In the light of the above observations from state-of-the-art, some clarifications are required on the different methodologies which are available in the literature. Instead of choosing a turnkey solution from an existing solver, the present work proposes to investigate and compare different promising methodologies. The major contribution of such an approach is to



(a) *Simulation from the DIVA solver* [4] **(b)** *Simulation from the ARCHER solver* [126]

Figure 5: *Two simulations of droplet evaporating in a turbulent flow. The left figure represent a simulation of a group of droplets with injected turbulence using a Level-Set methodology. The right figure present the evaporation of liquid in HIT using a Coupled LS-VOF methodology.*

give a better understanding of the numerical methods applied to phase-change simulations based on thorough numerical experiments.
For this aim, a solver needs to be developed to embrace all these numerical methods. The starting point is the "in-house" solver TITAN containing the basis for classic domain decomposition and parallel computing on cartesian meshes. In addition, the solver discretization relies on the finite-volume method. Then, the numerical methods associated with the resolution of the governing equations of two-phase flows remain to be determined and implemented, which constitutes the central part of the PhD work.
While VOF and LS are Eulerian methods that can rely on classic domain decomposition for parallel computation, FT is a Lagrangian class of methods that requires specific parallel algorithms. Therefore, the manuscript focuses on VOF and LS class of methods to concur with the target solver framework, which is only ready for cartesian domain decomposition.

Outline of the manuscript

The manuscript describes the pathway to set up an incompressible two-phase flow solver with phase change.
Part I describes the mathematical formulation for the derivation of two-phase flows in Chap-

ter 1 with the associated physical assumptions. In Chapter 2, a finite-volume derivation of the resulting set of governing equations is provided, leading to the VOF two-fluid and one-fluid formulations. The difficulty encountered when the same derivation is applied to LS is also detailed.
Then Part II gives a state-of-the-art of interface capturing methods in Chapter 3. From the literature, four methods are selected and compared in Chapter 4 based on shared metrics for accuracy and mass conservation.
Part III details the numerical challenges of two-phase flows without phase change with emphasis on the momentum conservation and the surface tension modelling in Chapter 5. Based on these considerations, a unified framework is presented and coupled to the four interface capturing methods evaluated before. For validation purposes, Chapter 6 presents comparisons of this unified framework adapted to the four interface capturing methods based on momentum conservation and curvature accuracy.
At this stage of the manuscript, the solver is able to simulate two-phase flows and Part IV addresses its extension to evaporation simulations. First, a state-of-the-art of the numerical challenges arising from the mass transfer at the interface is detailed in Chapter 7. The study concentrates on handling a velocity jump in the mass and momentum equations and the resolution of the energy and species mass fraction, which are crucial quantities to describe the phase-change process. Chapter 8 focuses on the accurate reconstruction of the quantities at the interface. This encompasses the evaporation rate, the interface temperature, and the species mass fraction at the interface. Finally, Chapter 9 presents an extension of the unified framework to phase-change with applications on planar, cylindrical, and spherical evaporation test cases.

Part I

Physical description of an interface

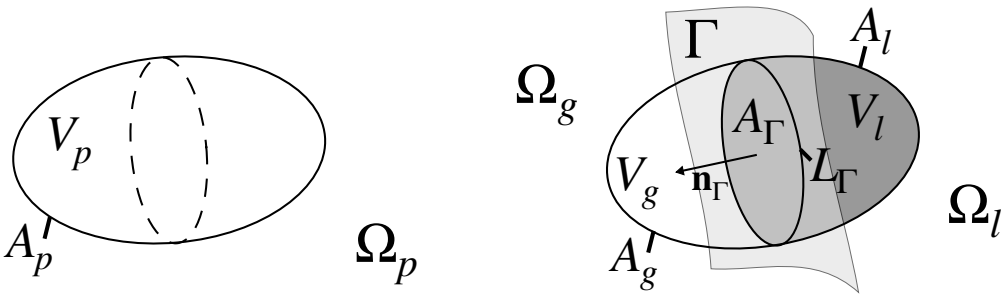
Chapter 1

Governing equations of two-phase flows

Contents

In the following, the governing equations of two-phase flows are presented essentially based on the derivation of Ishii & Hibiki [87]. In Section 1.1 the governing equations are first derived for a single-fluid particle illustrated in Fig. 1.1a leading to a local instantaneous formulation. When the fluid particle contains both phases as in Fig. 1.1b, the governing equations need to also include additional contributions due to the presence of the interface. These contributions can be written in terms of jump conditions at the interface, which are obtained from an interface balance. This derivation is provided in Section 1.2 based on the infinitely-thin or sharp interface assumption. Finally, Section 1.3 presents the hypothesis used in the present work leading to a simplified set of governing equations with the

associated jump conditions. The thermodynamic closures used for phase change are also derived.



(a) *Single-phase* **(b)** *Two-phase*

Figure 1.1: *Illustration of fluid particles in the single-phase and two-phase cases.* Ω_p, Ω_l *and* Ω_g *are the sub-domain corresponding to a single-phase flow, the gas phase and the liquid phase of a two-phase flow respectively.* V_p, V_l *and* V_g *are the material volumes delimited by material surfaces* A_p, A_l *and* A_g *corresponding to a general single-phase fluid particle, the liquid part and the gas part of the two-phase particle respectively.* A_Γ, L_Γ *and* $\mathbf{n}_\Gamma$ *are the surface, the length and the normal of the interface* Γ.

1.1 Local instantaneous formulation of single-phase flows

First, the governing equations are given for a fluid particle belonging to a single-phase flow. For a fluid particle belonging to one fluid only, as in Fig. 1.1a, the general integral balance of a quantity Φ in the material volume V_p delimited by its material surface A_p of the fluid can be written as

$$\frac{d}{dt}\int_{V_p}\rho\Phi dv = -\oint_{A_p}\mathbf{n}\cdot\mathbf{F}da + \int_{V_p}\rho S dv \quad , \tag{1.1}$$

with t the time, ρ the density of the fluid and $\mathbf{F}$ and S the fluxes and a volumetric source associated to Φ respectively.
This equation means that the change of $\rho\Phi$ in V_p is due to fluxes at the surface A_p and the volumetric source.
By using the Green's theorem and the Reynolds transport theorem, Eq. (1.1) can be rewritten into an instantaneous local balance of Φ

$$\boxed{\frac{\partial\rho\Phi}{\partial t} + \nabla\cdot(\mathbf{u}\rho\Phi) = -\nabla\cdot\mathbf{F} + \rho S \quad ,} \tag{1.2}$$

with $\mathbf{u}$ the fluid velocity.

1.1.1 Continuity equation

The continuity equation is obtained by replacing $\Phi = 1$, $\mathbf{F} = 0$ and $S = 0$ in Eq. (1.2):

$$\boxed{\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad .} \tag{1.3}$$

1.1.2 Momentum equation

The momentum equation corresponds to Eq. (1.2) with

$$\Phi = \mathbf{u} \quad , \tag{1.4a}$$
$$\mathbf{F} = -\mathcal{T} = P\mathbf{I} - 2\mu\mathbf{D} \quad , \tag{1.4b}$$
$$S = \mathbf{f}_v \quad . \tag{1.4c}$$

$\mathcal{T}$ is the stress tensor, P is the pressure in fluid, $\mathbf{I}$ is the identity matrix, $\mathbf{f}_v$ the body force, μ is the dynamic viscosity of the fluid and $\mathbf{D}$ is the rate-of-deformation tensor:

$$\mathbf{D} = \frac{1}{2}\left(\nabla \mathbf{u} + \nabla^{\intercal} \mathbf{u}\right) - \frac{1}{3}\left(\nabla \cdot \mathbf{u}\right)\mathbf{I} \quad . \tag{1.5}$$

Note that in this definition of $\mathcal{T}$, the fluid is considered Newtonian.
The local balance of momentum is then obtained

$$\boxed{\frac{\partial \rho \mathbf{u}}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla P + \nabla \cdot (2\mu \mathbf{D}) + \rho \mathbf{f}_v \quad .} \tag{1.6}$$

1.1.3 Energy equation

The total energy equation is obtained by choosing:

$$\Phi = e_i + \frac{1}{2}\left\|\mathbf{u}\right\|_2^2 \quad , \tag{1.7a}$$
$$\mathbf{F} = \mathbf{Q} - \mathcal{T} \cdot \mathbf{u} \quad , \tag{1.7b}$$
$$S = \mathbf{f}_v \cdot \mathbf{u} + \frac{\dot{q}}{\rho} \quad , \tag{1.7c}$$

with e_i the internal energy, $\mathbf{Q}$ the thermal fluxes and $\frac{\dot{q}}{\rho}$ the thermal source of the fluid. $\left\|\cdot\right\|_2$ stands for the L_2 norm.
This gives the following balance equation of total energy

$$\begin{aligned} \frac{\partial \rho \left(e_i + \frac{1}{2}\left\|\mathbf{u}\right\|_2^2\right)}{\partial t} + \nabla \cdot \left(\rho \left(e_i + \frac{1}{2}\left\|\mathbf{u}\right\|_2^2\right)\mathbf{u}\right) \\ = -\nabla \mathbf{Q} + \nabla \cdot (\mathcal{T} \cdot \mathbf{u}) + \rho \mathbf{f}_v \cdot \mathbf{u} + \dot{q} \quad . \end{aligned} \tag{1.8}$$

Now, an equation for the enthalpy is also provided as it corresponds to the energy-related quantity of interest in our simplified two-phase flow equations. For this purpose, the mechanical energy equation is given. It is obtained from multiplying Eq. (1.6) by $\mathbf{u}$:

$$\frac{\partial \rho \frac{1}{2} \|\mathbf{u}\|_2^2}{\partial t} + \nabla \cdot \left(\rho \frac{1}{2} \|\mathbf{u}\|_2^2 \mathbf{u} \right) = -\mathbf{u} \cdot \nabla P + (\nabla \cdot \mathcal{T}) + \rho \mathbf{f}_v \cdot \mathbf{u} \quad . \tag{1.9}$$

Then, by subtracting Eq. (1.9) to Eq. (1.8), the internal energy equation is retrieved

$$\frac{\partial \rho e_i}{\partial t} + \nabla \cdot (\rho e_i \mathbf{u}) = -\nabla \mathbf{Q} - \mathbf{u} \cdot \nabla P + \nabla \mathbf{u} : \mathcal{T} + \dot{q} \quad . \tag{1.10}$$

Finally, the enthalpy can be introduced as

$$h_t = e_i + \frac{P}{\rho} \quad , \tag{1.11}$$

which leads to the following enthalpy equation

$$\boxed{\frac{\partial \rho h_t}{\partial t} + \nabla \cdot (\rho h_t \mathbf{u}) = -\nabla \mathbf{Q} - \frac{DP}{Dt} + \nabla \mathbf{u} : \mathcal{T} + \dot{q} \quad .} \tag{1.12}$$

with $\frac{D}{Dt}$ the material derivative related to the fluid such that for a quantity Φ:

$$\frac{D\Phi}{Dt} = \frac{\partial \Phi}{\partial t} + \mathbf{u} \cdot \nabla \Phi \quad . \tag{1.13}$$

1.2 Interface balance

The balance equations presented in Section 1.1 can be applied to any fluid particle belonging to one of the fluids of the two-phase flow system. However, if a fluid particle is crossed by the interface as in Fig. 1.1b, additional terms appear in the balance equation accounting for the discontinuities at the interface. They take the form of jump conditions across the singular surface A_Γ transcribing the exchange of mass, momentum and energy at the interface.
The derivation of these jumps can be performed by considering the volume of interest V_I in Fig. 1.2. This volume is obtained by taking a slice of infinitesimal width $\delta = \delta_1 + \delta_2$ of the particle flow at the interface. Then V_I is delimited by the area A_1 and A_2 respectively in the gas and liquid phases with the outward pointing normals $\mathbf{n}_1$ and $\mathbf{n}_2$ and the surface Σ_I with the associated normal $\mathbf{n}_\Sigma$. The integral balance of Φ in the control volume V_I is then written as

$$\begin{aligned} \frac{d}{dt} \int_{V_I} \rho \Phi dv = &\sum_{p=1}^{2} \int_{A_p} \mathbf{n}_p \cdot \left((u_p - u_I) \rho \Phi + \mathbf{F}_p \right) da + \int_{V_I} \rho S dv \\ &- \int_{L_\Gamma} \int_{-\delta_2}^{\delta_1} \mathbf{n}_\Gamma \cdot \mathbf{F}_\Gamma d\delta dl \quad , \end{aligned} \tag{1.14}$$

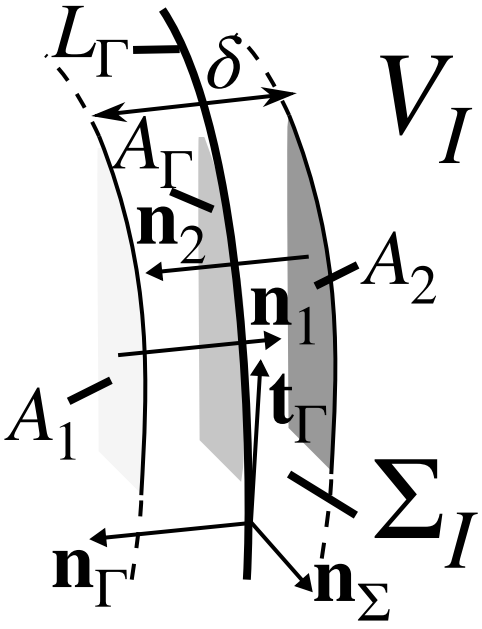


Figure 1.2: *Illustration of the interface balance: V_I is the volume of interest on which the balance is derived. A_1 and A_2 correspond to the boundary surfaces of V_I from the gas and liquid side respectively with their associated outward pointing normals $\mathbf{n}_1$ and $\mathbf{n}_2$. Σ_I is the other boundary surface of V_I sharing a liquid and a gas part of width $\delta = \delta_1 + \delta_2$ with the associated outward pointing normal $\mathbf{n}_\Sigma$. A_Γ and L_Γ is the interface and length of the interface with associated normal and tangential vectors $\mathbf{n}_\Gamma$ and $\mathbf{t}_\Gamma$ respectively.*

with $\mathbf{u}_I$ the velocity of the control volume V_I. Based on the sharp interface assumption of our framework, δ tends to zero. Then $\int_{-\delta_2}^{\delta_1} \mathbf{n}_\Sigma \cdot \mathbf{F}_\Gamma d\delta$ simplifies to $\mathbf{n}_\Sigma \cdot \mathbf{F}_\Gamma$ and $\int_{V_I} dv$ is reexpressed into the surface integral $\int_{A_\Gamma} da$. This leads to the following surface balance on A_Γ

$$\begin{aligned}\frac{d}{dt}\int_{A_\Gamma} \rho\Phi da = &\sum_{p=1}^{2} \int_{A_p} \mathbf{n}_p \cdot \left(\left(\mathbf{u}_p - \mathbf{u}_\Gamma\right)\rho\Phi + \mathbf{F}_p\right) da + \int_{A_\Gamma} \rho S da \\ &- \int_{L_\Gamma} \mathbf{n}_\Sigma \cdot \mathbf{F}_\Gamma dl \quad , \end{aligned} \tag{1.15}$$

By using the surface Green's theorem, the surface transport theorem and by noticing that A_1, A_2 and A_Γ coincide if δ is small enough, Eq. (1.15) can be rewritten into an instantaneous local balance

$$\frac{\partial \rho_\Gamma \Phi}{\partial t} + \rho_\Gamma \Phi \nabla \cdot \mathbf{u}_\Gamma = \sum_{p=1}^{2} \left(\rho_p \Phi_p \mathbf{n}_p \cdot \left(\mathbf{u}_p - \mathbf{u}_\Gamma\right) + \mathbf{n}_p \cdot \mathbf{F}_p\right) - \mathbf{n}_\Sigma \cdot \mathbf{F}_\Gamma + \rho_\Gamma S \quad . \tag{1.16}$$

Here, the mass density of interface $\rho_\Gamma = \rho\delta$ is introduced. In the present study, the interface is considered to be infinitely thin and massless with $\rho_\Gamma = 0$. This implies that

momentum and energy in the interface can also be neglected. Furthermore, the flux along the line L_Γ is supposed to be negligible too. All these assumptions lead to the local interface balance for a quantity Φ:

$$\sum_{p=1}^{2} \left(\rho_p \Phi_p \mathbf{n}_p \cdot (\mathbf{u}_p - \mathbf{u}_\Gamma) + \mathbf{n}_p \cdot \mathbf{F}_p\right) - \mathbf{n}_\Sigma \cdot \mathbf{F}_\Gamma = 0 \quad , \tag{1.17}$$

by introducing the jump operator $[\Phi]_\Gamma$ such that

$$[\Phi]_\Gamma = \Phi_l - \Phi_g \quad , \tag{1.18}$$

and the interface normal $\mathbf{n}_\Gamma$ defined by the outward pointing liquid normal

$$\mathbf{n}_\Gamma = \mathbf{n}_l = -\mathbf{n}_g \quad , \tag{1.19}$$

then Eq. (1.17) can be rewritten as

$$\boxed{[\rho\Phi\,(\mathbf{u} - \mathbf{u}_\Gamma) + \mathbf{F}]_\Gamma \cdot \mathbf{n}_\Gamma = \mathbf{n}_\Sigma \cdot \mathbf{F}_\Gamma \quad .} \tag{1.20}$$

1.2.1 Mass balance

The mass balance is obtained by replacing $\Phi = 1$, $\mathbf{F} = 0$ and $\mathbf{F}_\Gamma = 0$ in Eq. (1.20):

$$[\rho\,(\mathbf{u} - \mathbf{u}_\Gamma)]_\Gamma \cdot \mathbf{n}_\Gamma = 0 \quad . \tag{1.21}$$

This jump condition expresses the mass flux through the interface $\dot{m}$ such that

$$\dot{m} = \rho_l\,(\mathbf{u}_l - \mathbf{u}_\Gamma) = \rho_g\,(\mathbf{u}_g - \mathbf{u}_\Gamma) \quad . \tag{1.22}$$

In the present work, $\dot{m}$ is referred as the evaporation rate, and combining Eqs. (1.21) and (1.22) leads to the following velocity jump:

$$\boxed{[\mathbf{u}]_\Gamma \cdot \mathbf{n}_\Gamma = \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \quad .} \tag{1.23}$$

1.2.2 Momentum balance

The momentum balance corresponds to Eq. (1.20) with the definitions in Eq. (1.4) and $\mathbf{F}_\Gamma = \sigma\kappa\mathbf{n}_\Gamma + \nabla^s\sigma$ with ∇^s the surface gradient. The momentum balance at the interface is then obtained:

$$[\rho\mathbf{u} \otimes (\mathbf{u} - \mathbf{u}_\Gamma) - P\mathbf{I} + 2\mu\mathbf{D}]_\Gamma \cdot \mathbf{n}_\Gamma = \sigma\kappa\mathbf{n}_\Gamma + \nabla^s\sigma \quad , \tag{1.24}$$

which can be decomposed into the normal and tangential components

$$\boxed{\begin{aligned} &\dot{m}\,[\mathbf{u}]_\Gamma \cdot \mathbf{n}_\Gamma - [P]_\Gamma + 2\,[\mu\mathbf{D} \cdot \mathbf{n}_\Gamma]_\Gamma \cdot \mathbf{n}_\Gamma = \sigma\kappa \quad , \qquad (1.25)\\ &\dot{m}\,[\mathbf{u}]_\Gamma \cdot \mathbf{t}_\Gamma + 2\,[\mu\mathbf{D} \cdot \mathbf{n}_\Gamma]_\Gamma \cdot \mathbf{t}_\Gamma = \nabla^s\sigma \quad . \qquad (1.26)\end{aligned}}$$

1.2.3 Energy balance

The total energy balance corresponds to Eq. (1.20) with the definitions in Eq. (1.7) and $\mathbf{F}_\Gamma = \sigma\kappa\mathbf{n}_\Gamma \cdot \mathbf{u}_\Gamma$

$$\left[\rho\left(e_i + \frac{1}{2}\left\|\mathbf{u}\right\|_2^2\right)(\mathbf{u} - \mathbf{u}_\Gamma) - \mathcal{T}_p \cdot \mathbf{u} + \mathbf{Q}\right]_\Gamma \cdot \mathbf{n}_\Gamma = \sigma\kappa\mathbf{n}_\Gamma \cdot \mathbf{u}_\Gamma \quad . \tag{1.27}$$

As for the momentum jump condition, it is possible to reformulate some terms using $\dot{m}$

$$\dot{m}\left[e_i + \frac{1}{2}\left\|\mathbf{u}\right\|_2^2\right]_\Gamma + \left[\mathbf{Q}\right]_\Gamma \cdot \mathbf{n}_\Gamma - \left[\mathcal{T}_p \cdot \mathbf{u}\right]_\Gamma \cdot \mathbf{n}_\Gamma = \sigma\kappa\mathbf{n}_\Gamma \cdot \mathbf{u}_\Gamma \quad . \tag{1.28}$$

Finally, the enthalpy balance at the interface is obtained by first deriving the balance of the mechanical energy equation:

$$\dot{m}\left[\frac{1}{2}\left\|\mathbf{u}\right\|_2^2\right]_\Gamma - \left[\mathcal{T}_p \cdot \mathbf{u}\right]_\Gamma \cdot \mathbf{n}_\Gamma = \sigma\kappa\mathbf{n}_\Gamma \cdot \mathbf{u}_\Gamma \quad . \tag{1.29}$$

Then, by subtracting Eq. (1.29) to Eq. (1.28), the enthalpy jump is written as

$$\boxed{\dot{m}\left[h_t + \frac{P}{\rho}\right]_\Gamma + \left[\mathbf{Q}\right]_\Gamma \cdot \mathbf{n}_\Gamma = 0 \quad .} \tag{1.30}$$

1.2.4 Entropy production

Based on the work of Delhaye [44], the entropy variation at the interface is written as

$$\begin{aligned}\Delta s_\Gamma = &\sum_{p=1}^{2} \frac{\dot{m}_p}{T_\Gamma}\left(g_p + \frac{\left\|\mathbf{u}_p - \mathbf{u}_\Gamma\right\|_2^2}{2} - \frac{2\mu_p \mathbf{D}_p \cdot \mathbf{n}_p \cdot \mathbf{n}_p}{\rho_p}\right) \\ &- \sum_{p=1}^{2} \frac{2\mu_p \mathbf{D}_p \cdot \mathbf{n}_p \cdot \mathbf{t}_p}{T_\Gamma}\left(\mathbf{u}_p \cdot \mathbf{t}_p - \mathbf{u}_\Gamma \cdot \mathbf{t}_p\right) \\ &+ \sum_{p=1}^{2} \left(\mathbf{n}_p \cdot \mathbf{Q}_p + \dot{m}_p s_p T_p\right)\left(\frac{1}{T_\Gamma} - \frac{1}{T_p}\right) \quad ,\end{aligned} \tag{1.31}$$

with g the Gibbs free energy and s the entropy. Then, the entropy production at the interface is assumed to be zero. Moreover, the three terms are considered to be independently zero.

Chemical disequilibrium Cancellation of the first term leads to a chemical jump condition stating that the chemical potential difference is due to a mechanical disequilibrium at the interface

$$\left[g\right]_\Gamma = -\left[\frac{\left\|\mathbf{u} - \mathbf{u}_\Gamma\right\|_2^2}{2}\right]_\Gamma + \left[\frac{2\mu_p \mathbf{D}_p \cdot \mathbf{n}_\Gamma \cdot \mathbf{n}_\Gamma}{\rho}\right]_\Gamma \quad . \tag{1.32}$$

No-slip condition The second term states that there is no-slip between the two fluids at the interface

$$\mathbf{u}_l^\Gamma \cdot \mathbf{t}_\Gamma = \mathbf{u}_g^\Gamma \cdot \mathbf{t}_\Gamma = \mathbf{u}_\Gamma \cdot \mathbf{t}_\Gamma \quad , \tag{1.33}$$

with $\mathbf{u}_l^\Gamma$ and $\mathbf{u}_g^\Gamma$ the liquid and gas velocities at the interface.

Thermal boundary condition Finally, the last term leads to continuity of the temperature at the interface

$$T_l^\Gamma = T_g^\Gamma = T_\Gamma \quad , \tag{1.34}$$

with T_l^Γ and T_g^Γ the liquid and gas temperatures at the interface.

1.3 Simplified two-phase flow equations

In the present work, the sharp interface representation is supplemented with several hypotheses on the two-phase system composed of two fluids

(i) Both fluids are supposed to be incompressible as the system is considered quasi-isobaric.
(ii) The first fluid is a pure mono-component liquid phase.
(iii) The other fluid consists of the vapour of the condensable species and inert gas.
(iv) The thermal exchanges are modelled with Fourier's law.
(v) The species diffusion is modelled with Fick's law.
(vi) All fluid properties are considered to be constant in the fluids. This includes ρ, μ, c_p, and k.
(vii) No Marangoni effects are considered here, hence σ is considered constant.
(viii) The specific latent heat of boiling $\mathscr{L}_{vap}$ does not vary with temperature.

By using this set of hypotheses, the governing equations used in this work are provided.

1.3.1 Mass conservation

In the incompressible form (i), the continuity equation Eq. (1.3) results in a simple divergence-free condition

$$\boxed{\nabla \cdot \mathbf{u} = 0 \quad ,} \tag{1.35}$$

with the velocity jump normal to the interface

$$\boxed{[\mathbf{u}]_\Gamma \cdot \mathbf{n}_\Gamma = \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \quad .} \tag{1.36}$$

1.3.2 Momentum conservation

From the conservation of momentum Eq. (1.6), by considering ρ and μ constant (vi), the following equation is retrieved

$$\boxed{\rho\left(\frac{\partial \mathbf{u}}{\partial t} + \nabla\cdot(\mathbf{u}\otimes\mathbf{u})\right) = -\nabla P + \nabla\cdot(2\mu\mathbf{D}) + \rho\mathbf{f}_v \quad .} \tag{1.37}$$

Furthermore, Eq. (1.35) can be injected in Eq. (1.5) to simplify the rate-of-deformation tensor

$$\mathbf{D} = \frac{1}{2}\left(\nabla\mathbf{u} + \nabla^{\intercal}\mathbf{u}\right) \quad , \tag{1.38}$$

The associated momentum jump conditions, in absence of Marangoni effects (vii) are:

$$\boxed{\begin{aligned} &\dot{m}\left[\mathbf{u}\right]_\Gamma\cdot\mathbf{n}_\Gamma - \left[P\right]_\Gamma + 2\left[\mu\mathbf{D}\cdot\mathbf{n}_\Gamma\right]_\Gamma\cdot\mathbf{n}_\Gamma = \sigma\kappa \quad , &(1.39)\\ &\dot{m}\left[\mathbf{u}\right]_\Gamma\cdot\mathbf{t}_\Gamma + 2\left[\mu\mathbf{D}\cdot\mathbf{n}_\Gamma\right]_\Gamma\cdot\mathbf{t}_\Gamma = 0 \quad . &(1.40)\end{aligned}}$$

1.3.3 Energy conservation

In an incompressible flow (i), the total energy of the system only depends on T and the enthalpy defined in Eq. (1.11) can be simplified to

$$h_t = c_p T \quad , \tag{1.41}$$

with c_p the specific heat at constant pressure assumed constant (vi).
Then, without source of heat, the energy conservation is expressed as

$$\frac{\partial \rho c_p T}{\partial t} + \nabla\cdot(\rho\mathbf{u}c_pT) = \nabla\mathbf{u} : (2\mu\mathbf{D}) - \nabla\cdot\mathbf{Q} \quad , \tag{1.42}$$

with $\nabla\mathbf{u} : (2\mu\mathbf{D})$ the viscous energy dissipation usually neglected and $\mathbf{Q}$ the heat flux following a Fourier's law (iv):

$$\mathbf{Q} = -k\nabla T \quad , \tag{1.43}$$

with k the thermal conductivity.
Finally, the thermal energy Eq. (1.42) can be rewritten as

$$\boxed{\rho c_p\left(\frac{\partial T}{\partial t} + \mathbf{u}\cdot\nabla T\right) = \nabla\cdot(k\nabla T) \quad .} \tag{1.44}$$

Accordingly, the jump of enthalpy introduced in Eq. (1.30) can be written in term of a temperature gradient jump using Eq. (1.43) and neglecting $\left[\frac{P}{\rho}\right]_\Gamma$:

$$\boxed{\left[k\nabla T\cdot\mathbf{n}_\Gamma\right]_\Gamma = \dot{m}\left[h_t\right]_\Gamma \quad ,} \tag{1.45}$$

with $[h_t]_\Gamma$ the enthalpy jump which contains a term due to the specific latent heat of boiling $\mathscr{L}_{vap}$ and another term due to the gap between the interface temperature T_Γ and the saturation temperature T_{sat} of the pure liquid (ii)

$$[h_t]_\Gamma = -\mathscr{L}_{vap} + [c_p]_\Gamma \left(T_\Gamma - T_{sat}\right) \quad . \tag{1.46}$$

1.3.4 Species mass fraction conservation

In the system, the gas is supposed to be bi-component with inert gas and the vapour of the liquid fluid (iii). Then, an equation is required to describe the mixture in the gas phase. The liquid vapour mass conservation can be deduced from Eq. (1.2) with $\Phi_p = Y$, $\mathbf{F}_p = \rho \mathbf{u}_v Y$ and $S_p = 0$

$$\frac{\partial \rho Y}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y) = -\nabla \cdot (\rho \mathbf{u}_v Y) \quad , \tag{1.47}$$

with $\mathbf{u}_v$ the diffusion velocity of the pure liquid vapour in the gas phase. It can be written in term of a diffusive flux for the vapour mass fraction as $\mathbf{J} = \rho \mathbf{u}_v Y$. Equation (1.47) is then rewritten as

$$\frac{\partial \rho Y}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y) = -\nabla \cdot \mathbf{J} \quad , \tag{1.48}$$

with $\mathbf{J}$ modelled using a Fick's law (v). In a bi-component mixture, it is written as

$$\mathbf{J} = -\rho \mathcal{D}_v \nabla Y \quad , \tag{1.49}$$

with $\mathcal{D}_v$ the vapour diffusivity in the gas.
Using Eq. (1.48) and the definition of $\mathbf{J}$ in Eq. (1.49) , the conservation of liquid vapour is defined by the following equation

$$\boxed{\rho \left(\frac{\partial Y}{\partial t} + \mathbf{u} \cdot \nabla Y \right) = \nabla \cdot (\rho \mathcal{D}_v \nabla Y) \quad .} \tag{1.50}$$

Note that Eq. (1.50) is only defined in the gas phase as the liquid is mono-component (ii) and the inert gas is insoluble in the liquid. This gives a simple relation for the inert gas mass fraction $Y_0 = 1 - Y$.
Integration of Eq. (1.48) leads to the following jump at the interface

$$\dot{m} \, [Y]_\Gamma + [\mathbf{J} \cdot \mathbf{n}_\Gamma]_\Gamma = 0 \quad . \tag{1.51}$$

As liquid is mono-component, $Y_l = 1$ and $\nabla Y|_l = 0$, the jump can be rewritten in term of vapour mass fraction gradient $\nabla Y|_\Gamma$ and vapour mass fraction value Y_Γ at the interface:

$$\boxed{\rho_g \mathcal{D}_v \, \nabla Y|_\Gamma \, \mathbf{n}_\Gamma = \dot{m} \left(1 - Y_\Gamma\right) \quad .} \tag{1.52}$$

1.3.5 Thermodynamic closure at the interface

Based on the above hypotheses, the phase change at the interface can be described using thermodynamic closures.
First, the evaporation rate $\dot{m}$ can be defined either using the heat flux jump defined by Eq. (1.45) or the mass flux jump defined by Eq. (1.52)

$$\boxed{\dot{m} = \frac{[k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma}{[h_t]_\Gamma} \quad ,} \tag{1.53}$$

$$\boxed{\dot{m} = \frac{\rho_g \mathcal{D}_v \nabla Y \mathbf{n}_\Gamma}{Y_\Gamma - 1} \quad ,} \tag{1.54}$$

both simultaneously true at the interface location.
The interface quantities at the interface T_Γ and Y_Γ need to be defined. Assuming that the interface is at thermodynamic equilibrium, pressure, temperature and chemical potential are equal in the two phases. Any changes in chemical potential $d\mu_p^p = v_k dP_s - s_p dT$ is also equal along the interface. Here, P_s is the saturation vapour pressure at the interface while v_p and s_p are the specific volume and entropy of the phase respectively. This corresponds to the two equalities:

$$d\mu_l^p = d\mu_g^p \quad , \tag{1.55}$$

$$v_l dP_s - s_l dT = v_g dP_s - s_g dT \quad , \tag{1.56}$$

which gives a relation between P_s and T at the interface

$$\left(\frac{dP_s}{dT}\right) = \frac{[s]_\Gamma}{[v]_\Gamma} \quad , \tag{1.57}$$

by noticing that v_l is much lower than v_g for a liquid-gas system, one can approximate $[v]_\Gamma = -v_g$. Furthermore, the specific volume of gas can be related to temperature and pressure by using the perfect gaz hypothesis $Pv_g = RT$ with R the universal gas constant. Also, the jump in entropy at the interface is related to the specific latent heat of boiling $\mathscr{L}_{vap}$ through $T_\Gamma [s]_\Gamma = -\mathscr{L}_{vap}$. From those observations and Eq. (1.57), the Clausius-Clayperon relation is obtained

$$\left(\frac{dP_s}{dT}\right) = \frac{P_s \mathscr{L}_{vap}}{R T_\Gamma^2} \quad . \tag{1.58}$$

Eq. (1.58) can be reexpressed by using $dP_s = P_s d\ln P_s$:

$$\left(\frac{d\ln P_s}{dT}\right) = \frac{\mathscr{L}_{vap}}{R T_\Gamma^2} \quad . \tag{1.59}$$

Finally, the Rankine relation is obtained by integration of Eq. (1.59) from a reference state, taken here as T_{sat} at a reference pressure P_{ref}

$$\boxed{P_s = P_{ref} \exp\left(-\frac{\mathscr{L}_{vap}}{R}\left(\frac{1}{T_\Gamma} - \frac{1}{T_{sat}}\right)\right) \quad .} \tag{1.60}$$

In this integration, $\mathscr{L}_{vap}$ and R are supposed constant with temperature (viii), which is not always true (see [98] for some extended studies). Note that some authors use different relations such as in FS3D [195; 179] for which authors prefer the Wagner equation [177]. Dodd et al. [52] and Bures et al. [24] use the Antoine equation [6] instead. These other formulations rely on empirical corrections to take into account dependencies of $\mathscr{L}_{vap}$ with the temperature.
Now, a relation has to be established between the liquid vapour mass fraction Y and the saturation vapour pressure P_s. This is done using the definition of Y in terms of mole quantity n and molar mass M of both vapour and inert gas

$$Y = \frac{n_v M_v}{n_v M_v + n_0 M_0} \quad , \tag{1.61}$$

using the perfect gas relation to the partial pressures $P_v V = n_v RT$ this gives

$$Y = \frac{P_v M_v}{P_v M_v + P_0 M_0} \quad . \tag{1.62}$$

In a perfect gas, the total pressure is the sum of partial pressures $P_{ref} = P_v + P_0$ which leads to

$$Y = \frac{P_v M_v}{P_v \left(M_v - M_0\right) + P_{ref} M_0} \quad . \tag{1.63}$$

Then applying Raoult's law at the interface, the partial pressure is related to the saturation vapour pressure and liquid molar fraction X_v through $P_v = X_v P_s$. In a mono-component liquid, $X_v = 1$ and Raoult's law is reduced to $P_v = P_s$. The final form definition of Y_Γ is then

$$\boxed{Y_\Gamma = \frac{P_s M_v}{P_s M_v + \left(P_{ref} - P_s\right) M_0} \quad .} \tag{1.64}$$

Using Eqs. (1.60) and (1.64) gives a direct relation between T_Γ and Y_Γ.

1.4 Conclusion

This first chapter presented the simplified two-phase flow equations considered in the remainder of the manuscript. It contains the local instantaneous balance of mass, momentum, energy and mass species fractions with associated jump conditions at the interface, which have to be taken into account to correctly describe the two-phase system. In Part III, a first investigation of two-phase flows is provided without including phase change. The equation system is reduced to

$$\boxed{\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \quad , &(1.65a)\\ &\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) \right) = -\nabla P + \nabla \cdot (2\mu \mathbf{D}) + \rho \mathbf{f}_v \quad , &(1.65b) \end{aligned}}$$

with the associated jumps

$$\boxed{\begin{aligned} &[\mathbf{u}]_\Gamma \cdot \mathbf{n}_\Gamma = 0 \quad , && (1.66a)\\ &[P]_\Gamma - 2\,[\mu \mathbf{D} \cdot \mathbf{n}_\Gamma]_\Gamma \cdot \mathbf{n}_\Gamma = -\sigma\kappa \quad . && (1.66b) \end{aligned}}$$

In Part IV, evaporation is included in the equation system leading to

$$\boxed{\begin{aligned} &\nabla \cdot \mathbf{u} = 0 \quad , && (1.67a)\\ &\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \nabla \cdot (\mathbf{u} \otimes \mathbf{u}) \right) = -\nabla P + \nabla \cdot (2\mu \mathbf{D}) + \rho \mathbf{f}_v \quad , && (1.67b)\\ &\rho c_p \left(\frac{\partial T}{\partial t} + \mathbf{u} \cdot \nabla T \right) = \nabla \cdot (k \nabla T) \quad , && (1.67c)\\ &\rho \left(\frac{\partial Y}{\partial t} + \mathbf{u} \cdot \nabla Y \right) = \nabla \cdot (\rho \mathcal{D}_v \nabla Y) \quad , && (1.67d) \end{aligned}}$$

and

$$\boxed{\begin{aligned} &[\mathbf{u}]_\Gamma \cdot \mathbf{n}_\Gamma = \dot{m} \left[\frac{1}{\rho} \right]_\Gamma \quad , && (1.68a)\\ &\dot{m}\,[\mathbf{u}]_\Gamma \cdot \mathbf{n}_\Gamma - [P]_\Gamma + 2\,[\mu \mathbf{D} \cdot \mathbf{n}_\Gamma]_\Gamma \cdot \mathbf{n}_\Gamma = \sigma\kappa \quad , && (1.68b)\\ &\dot{m}\,[\mathbf{u}]_\Gamma \cdot \mathbf{t}_\Gamma + 2\,[\mu \mathbf{D} \cdot \mathbf{n}_\Gamma]_\Gamma \cdot \mathbf{t}_\Gamma = 0 \quad , && (1.68c)\\ &[k \nabla T \cdot \mathbf{n}_\Gamma]_\Gamma = \dot{m}\,[h_t]_\Gamma \quad , && (1.68d)\\ &\rho_g \mathcal{D}_v \left. \nabla Y \right|_\Gamma \mathbf{n}_\Gamma = \dot{m}\,(1 - Y_\Gamma) \quad . && (1.68e) \end{aligned}}$$

The next chapter covers the derivation of the system of equations resulting from the combination of local balance and jump conditions at the interface.

Chapter 2

Toward the numerical resolution of two-phase flow equations

Contents

Now that the governing equations of incompressible two-phase flows have been provided, further derivations are required to close the system of equations. More specifically, the governing equations valid for the phases taken separately need to be bridged by imposing the jump conditions correctly at the interface. For this purpose, a discrete set of equations can be established from the derivation of one-fluid or two-fluid formulations, which both rely on the explicit description of the interface position through an indicator function χ_p. This chapter aims to derive these equations properly, either using Volume-Of-Fluid or Level-Set representations of the interface.

In Section 2.1, the definition of χ_p is provided with the associated manners to transport it. Then, the derivation of one-fluid and two-fluid equations using a finite-volume discretization is detailed, which naturally leads to the geometric VOF representation of the interface in Section 2.3. The use of geometric VOF can be complex to solve numerically and Section 2.4 explores the alternative derivations which can be used to obtain one-fluid and two-fluid formulations based on strong assumptions. Then, Section 2.5 contains an open discussion on the general derivation of a one-fluid or two-fluid set of equations that could be applied to Level-Set methods. Finally, some clarifications on the different formulations are provided in light of the present framework in Section 2.6.

2.1 Indicator function and volume fraction

The indicator function χ_p is defined such that

$$\chi_p(\mathbf{x}) = \begin{cases} 1 & \text{if } \mathbf{x} \in \Omega_p \\ 0 & \text{otherwise} \end{cases} \quad , \tag{2.1}$$

where Ω_p is the domain containing the fluid p and $\mathbf{x}$ is the spatial coordinate. This function evolves through the general transport equation

$$\frac{\partial \chi_p}{\partial t} + \mathbf{u}_\Gamma \cdot \nabla \chi_p = 0 \quad , \tag{2.2}$$

with $\mathbf{u}_\Gamma$ the interface velocity (which needs a closure).
As χ_p is intrinsically discontinuous, solving Eq. (2.2) numerically is challenging ,and conventional discretizations would lead to oscillations or smearing of the sharp discontinuity. Four solutions can be envisaged to solve Eq. (2.2):

- Keeping the sharp nature of χ_p in the context of finite-volume discretizations: the Geometric Volume-of-Fluid methods [148; 181; 168]
- Accepting to lose the sharp nature of χ_p, but using compressive and anti-diffusive numerical schemes to keep maintain it as sharp as possible: the Algebraic Volume-of-Fluid (VOF) methods [231; 84; 200]
- Replacing χ_p by a function which is regular enough to employ conventional discretizations for its transport: the Level-Set methods [153; 214; 150; 48]
- Averaging χ_p to smooth it enough such as multifluid [191; 37]: the Averaged methods.

Depending on the strategy, χ_p is replaced by another transported function, for which meaning and origin have to be clearly stated to close the equations.

2.1.1 Geometric VOF procedure

Let us first consider the geometric VOF method. In this strategy, the volume fraction is transported in each numerical cell and the average of finite-volume methods can be defined in a control volume $V_\mathcal{C}$ as

$$\alpha_p = \frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} \chi_p dv = \frac{V_{\mathcal{C},p}}{V_\mathcal{C}} \quad , \tag{2.3}$$

with $V_{\mathcal{C},p}$ the control volume of the phase p contained in $V_\mathcal{C}$. In this sense, the VOF method is a straight discretization of Eq. (2.2), and the semi-discrete equation is directly obtained by integrating Eq. (2.2) over each computational cell:

$$\frac{\partial \alpha_p}{\partial t} = -\frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} \mathbf{u}_\Gamma \cdot \nabla \chi_p dv = -\frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} \mathbf{u} \cdot \nabla \chi_p dv + \frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} (\mathbf{u} - \mathbf{u}_\Gamma) \cdot \nabla \chi_p dv \tag{2.4}$$

$$= -\frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} \nabla \cdot (\mathbf{u}\chi_p) dv + \frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} \chi_p \nabla \cdot \mathbf{u} dv + \frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} (\mathbf{u} - \mathbf{u}_\Gamma) \cdot \nabla \chi_p dv \quad . \tag{2.5}$$

Considering that:

$$\frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} (\mathbf{u} - \mathbf{u}_\Gamma) \cdot \nabla \chi_p dv = \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left(\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma\right) \cdot \mathbf{n}_p da \quad , \tag{2.6}$$

where $\mathbf{u}_p^\Gamma$ is the local velocity of phase p next to the interface.
Under the divergence-free condition Eq. (1.35), the following transport equation is obtained:

$$\frac{\partial \alpha_p}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \chi_p \mathbf{u} \cdot \mathbf{n} da = \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left(\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma\right) \cdot \mathbf{n}_p da \quad . \tag{2.7}$$

Therefore, there is no continuous VOF equation. It is by essence the discretized equation of χ_p which still appears in Eq. (2.7) and is inherently defined from the explicit reconstruction of the interface provided by the geometric VOF.

2.1.2 Level-Set procedure

For Level-Set, χ_p is often replaced by a distance function ϕ defined as

$$\phi(\mathbf{x}) = \begin{cases} -\min\limits_{\forall \mathbf{x}_\Gamma \in \Gamma} |\mathbf{x}_\Gamma - \mathbf{x}| & \text{if } \mathbf{x} \in \Omega_g \\ \min\limits_{\forall \mathbf{x}_\Gamma \in \Gamma} |\mathbf{x}_\Gamma - \mathbf{x}| & \text{if } \mathbf{x} \in \Omega_l \\ 0 & \text{if } \mathbf{x} \in \Gamma \end{cases} , \tag{2.8}$$

where $\mathbf{x}_\Gamma$ is the interface location.
The signed distance ϕ is transported as

$$\frac{\partial \phi}{\partial t} + \mathbf{u}_\Gamma \cdot \nabla \phi = 0 \quad . \tag{2.9}$$

This quantity is defined everywhere, and thus, the transport velocity $\mathbf{u}_\Gamma$ must be extended in the entire domain. The most important aspect is that a specific value (the isocontour) of this function must represent the interface location and move at the interface velocity $\mathbf{u}_\Gamma$. The Level-Set leads to a standard advection equation which is straightforward to solve at the cost of losing the link with Eq. (2.2).

2.1.3 Averaging procedure

Finally, an averaging procedure can be used on χ_p to get:

$$\alpha_p = \langle \chi_p \rangle \quad . \tag{2.10}$$

The volume fraction is thus directly linked to the choice of the averaging operator. For example, an ensemble-averaging would lead to a volume fraction that is a probability of having one phase or another, and not a coexistence of phases. If a filtering operator is chosen, α_p represents the volume fraction in the vicinity (the filter size) of the point of interest. The averaging procedure (considering averaging operators that commute with time and space derivatives) applied to the transport equation of χ_p leads to:

$$\frac{\partial \alpha_p}{\partial t} + \nabla \cdot \mathbf{u}_p \alpha_p = \langle (\mathbf{u} - \mathbf{u}_\Gamma) \cdot \nabla \chi_p \rangle + \langle \chi_p \nabla \cdot \mathbf{u} \rangle \quad , \tag{2.11}$$

where $\alpha_p \mathbf{u}_p = \langle \chi_p \mathbf{u} \rangle$. The choice of the average operator is needed to close the term $\langle (\mathbf{u}_p - \mathbf{u}_\Gamma) \cdot \nabla \chi_p \rangle$ related to exchanges at the interface.
Note that giving up the sharp nature of VOF using an algebraic approach also results in averaging the equations as Eq. (2.3) is not true anymore.

2.2 General two-fluid transport equation

The general methodology to obtain a complete set of equations for two-phase flows is as follows. The two-fluid system of equations can be derived by following the work of Drew [53] which was originally applied to dispersed two-phase flows from an Eulerian perspective. For convenience, some mathematical operators are introduced:

- The time derivative

$$\chi_p \frac{\partial \rho \Phi}{\partial t} = \frac{\partial \chi_p \rho \Phi}{\partial t} - \rho \Phi \frac{\partial \chi_p}{\partial t} = \frac{\partial \chi_p \rho \Phi}{\partial t} + \rho \Phi \mathbf{u}_\Gamma \cdot \nabla \chi_p \quad , \tag{2.12}$$

- The spatial derivative

$$\chi_p \nabla \cdot (\mathbf{u} \rho \Phi) = \nabla \cdot (\chi_p \mathbf{u} \rho \Phi) - \rho \Phi \mathbf{u} \cdot \nabla \chi_p \quad , \tag{2.13}$$

- The flux divergence

$$\chi_p \nabla \cdot \mathbf{F} = \nabla \cdot (\chi_p \mathbf{F}) - \mathbf{F} \cdot \nabla \chi_p \quad . \tag{2.14}$$

Then the two-fluid version of the instantaneous local balance Eq. (1.2) can be written as

$$\boxed{\begin{aligned} \frac{\partial \chi_p \rho \Phi}{\partial t} + \nabla \cdot (\chi_p \mathbf{u} \rho \Phi) = -\nabla \cdot (\chi_p \mathbf{F}) + \chi_p S \\ + (\rho_p \Phi (\mathbf{u}_p - \mathbf{u}_\Gamma) + \mathbf{F}) \cdot \nabla \chi_p \quad . \end{aligned}} \tag{2.15}$$

The additional terms appearing on the right-hand side are the interface contributions related to the mass exchange and the flux of Φ between the phases, respectively.
This general equation can then be discretized or averaged to simplify its resolution.

2.3 Finite-volume discretization: the Volume-of-Fluid method

In this section, the derivation of the two-fluid and one-fluid formulations are detailed when using the finite-volume discretization described in Section 2.1.1.
Let us define the phase mass-averaged quantities:

$$\rho_p \alpha_p \Phi_p^m = \frac{1}{V_\mathcal{C}} \int_{V_\mathcal{C}} \chi_p \rho \Phi dv \quad . \tag{2.16}$$

By integrating Eq. (2.15) over the cell volume, the two-fluid transport of a quantity Φ is written as:

$$\boxed{\begin{aligned} \frac{\partial \rho_p \alpha_p \Phi_p^m}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \chi_p \rho \Phi \mathbf{u} \cdot \mathbf{n} da = -\frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \chi_p \mathbf{F} \cdot \mathbf{n} da + \rho_p \alpha_p S_p^m \\ + \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left(\rho \Phi (\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma) + \mathbf{F} \right) \cdot \mathbf{n}_p da \quad , \end{aligned}} \tag{2.17}$$

As a consequence of the presence of χ_p in the surface integrals, the local values of all quantities are required to close the equations even if cell-averaged quantities are transported. Thus, a formal continuous expression cannot be derived for the volume fraction and needs to stay in this discretized form. Starting from the general expression Eq. (2.17), the two-fluid and one-fluid VOF equations can be derived.

2.3.1 VOF two-fluid equations

The governing equations presented in Section 1.3 are derived by substituting Φ_p^m in Eq. (2.17) with the appropriate quantities.

2.3.1.1 Two-fluid continuity equation

$$\boxed{\frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \chi_p \mathbf{u} \cdot \mathbf{n} da = \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \mathbf{u}_p^\Gamma \cdot \mathbf{n}_p da \quad .} \tag{2.18}$$

2.3.1.2 Two-fluid momentum equation

$$\begin{aligned}\frac{\partial \rho_p \alpha_p \mathbf{u}_p^m}{\partial t} + \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \chi_p \rho \mathbf{u} \otimes \mathbf{u} \cdot \mathbf{n} da = & -\frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \chi_p P \mathbf{I} \cdot \mathbf{n} da \\ & + \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} 2\chi_p \mu \mathbf{D} \cdot \mathbf{n} da + \rho_p \alpha_p \mathbf{f}_v^m \\ & + \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \rho_p \mathbf{u}_p \otimes \left(\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma\right) \cdot \mathbf{n}_p da \\ & + \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \left(-P_p \mathbf{I} + 2\mu_p \mathbf{D}_p\right) \cdot \mathbf{n}_p da \quad . \end{aligned} \tag{2.19}$$

2.3.1.3 Two-fluid energy equation

$$\begin{aligned}\frac{\partial \rho_p \alpha_p (c_p T)_p^m}{\partial t} + \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \chi_p \rho c_p T \mathbf{u} \cdot \mathbf{n} da = & \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \chi_p k \nabla T \cdot \mathbf{n} da \\ & + \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \rho c_{p,p} T_p \left(\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma\right) \cdot \mathbf{n}_p da \\ & - \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} k_p \nabla T_p \cdot \mathbf{n}_p da \quad . \end{aligned} \tag{2.20}$$

2.3.1.4 Two-fluid species mass fraction equation

$$\begin{aligned}\frac{\partial \rho_p \alpha_p Y_p^m}{\partial t} + \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \chi_p \rho Y \mathbf{u} \cdot \mathbf{n} da = & \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \chi_p \mathcal{D}_v \nabla Y \cdot \mathbf{n} da \\ & + \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \rho Y (\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma) \cdot \mathbf{n}_p da \\ & - \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \mathcal{D} \nabla Y \cdot \mathbf{n}_p da \quad . \end{aligned} \tag{2.21}$$

2.3.2 VOF one-fluid equations

The one-fluid derivation is obtained by summing the two-fluid formulation for each phase. Summation leads to new averaged quantities Φ^V, Φ^M and Φ^H the volume, mass and en-

thalpy averages defined as in [122]

$$\Phi_p^V = \alpha_l \Phi_l + \alpha_g \Phi_g \quad , \tag{2.22a}$$

$$\Phi_p^M = \frac{\alpha_l \rho_l \Phi_l + \alpha_g \rho_g \Phi_g}{\rho^V} \quad , \tag{2.22b}$$

$$\Phi_p^H = \frac{\alpha_l \rho_l c_{p,l} \Phi_l + \alpha_g \rho_g c_{p,g} \Phi_g}{\rho^V c_p^M} \quad . \tag{2.22c}$$

By considering the VOF formulation, $\alpha_l = f$ and $\alpha_g = 1 - f$ and the averages can be rewritten as

$$\Phi_p^V = \Phi_g + f \left[\Phi\right]_\Gamma \quad , \tag{2.23a}$$

$$\Phi_p^M = \frac{\rho_g \Phi_g + f \left[\rho \Phi\right]_\Gamma}{\rho^V} \quad , \tag{2.23b}$$

$$\Phi_p^H = \frac{\rho_g c_{p,g} \Phi_g + f \left[\rho c_p \Phi\right]_\Gamma}{\rho^V c_p^M} \quad . \tag{2.23c}$$

2.3.2.1 One-fluid continuity equation

By summing Eq. (2.18) for $p = l, g$, the following expression is obtained

$$\boxed{\frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \mathbf{u} \cdot \mathbf{n} da = -\frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left[\mathbf{u}\right]_\Gamma \cdot \mathbf{n}_\Gamma da \quad .} \tag{2.24}$$

2.3.2.2 One-fluid momentum equation

By summing Eq. (2.19) for $p = l, g$ the following equation is retrieved

$$\begin{aligned} \frac{\partial \rho^V \mathbf{u}^M}{\partial t} + \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} \rho \mathbf{u} \otimes \mathbf{u} \cdot \mathbf{n} da = & -\frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} P \mathbf{I} \cdot \mathbf{n} da \\ & + \frac{1}{V_\mathcal{C}} \int\limits_{\partial V_\mathcal{C}} 2\mu \mathbf{D} \cdot \mathbf{n} da + \rho^V \mathbf{f}_v \\ & - \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \left[\rho \mathbf{u} \otimes (\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma)\right]_\Gamma \cdot \mathbf{n}_\Gamma da \\ & - \frac{1}{V_\mathcal{C}} \int\limits_{A_\Gamma} \left(-\left[P\right]_\Gamma \mathbf{I} + 2\left[\mu \mathbf{D}\right]_\Gamma\right) \cdot \mathbf{n}_\Gamma da \quad . \end{aligned} \tag{2.25}$$

Finally, the interface contribution can be substituted by the surface tension using Eq. (1.24)

$$\boxed{\begin{aligned}\frac{\partial \rho^V \mathbf{u}^M}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \rho \mathbf{u} \otimes \mathbf{u} \cdot \mathbf{n} da = & -\frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} P\mathbf{I} \cdot \mathbf{n} da \\ & + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} 2\mu \mathbf{D} \cdot \mathbf{n} da + \rho^V \mathbf{f}_v \\ & + \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \sigma \kappa \cdot \mathbf{n}_\Gamma da \quad .\end{aligned}} \tag{2.26}$$

2.3.2.3 One-fluid energy equation

By summing Eq. (2.20) for $p = l, g$, the following expression is obtained:

$$\begin{aligned}\frac{\partial \rho^V c_p^M T^H}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \rho c_p T \mathbf{u} \cdot \mathbf{n} da = & \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} k \nabla T \cdot \mathbf{n} da \\ & + \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left[\rho c_p T \left(\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma\right)\right]_\Gamma \cdot \mathbf{n}_p da \\ & - \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left[k \nabla T\right]_\Gamma \cdot \mathbf{n}_p da \quad .\end{aligned} \tag{2.27}$$

Finally, by substituting the flux jump with the expression given in Eq. (1.45)

$$\boxed{\begin{aligned}\frac{\partial \rho^V c_p^M T^H}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \rho c_p T \mathbf{u} \cdot \mathbf{n} da = & \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} k \nabla T \cdot \mathbf{n} da \\ & - \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left(\dot{m} \left(\mathscr{L}_{vap} + \left[c_p\right]_\Gamma T_{sat}\right)\right) da \quad .\end{aligned}} \tag{2.28}$$

2.3.2.4 One-fluid species mass fraction equation

By analogy to the energy equation, the sum Eq. (2.21) for $p = l, g$ gives

$$\begin{aligned}\frac{\partial \rho^V Y^M}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \rho Y \mathbf{u} \cdot \mathbf{n} da =& \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \mathcal{D} \nabla Y \cdot \mathbf{n} da \\ &+ \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left[\rho Y \left(\mathbf{u}_p^\Gamma - \mathbf{u}_\Gamma\right)\right]_\Gamma \cdot \mathbf{n}_p da \\ &- \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \left[\mathcal{D} \nabla Y\right]_\Gamma \cdot \mathbf{n}_p da \quad . \end{aligned} \tag{2.29}$$

By noticing that the interface contribution can be substituted by $\dot{m}$, the one-fluid species mass fraction equation is retrieved

$$\boxed{\begin{aligned}\frac{\partial \rho^V Y^M}{\partial t} + \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \rho Y \mathbf{u} \cdot \mathbf{n} da = & \frac{1}{V_\mathcal{C}} \int_{\partial V_\mathcal{C}} \mathcal{D} \nabla Y \cdot \mathbf{n} da \\ &- \frac{1}{V_\mathcal{C}} \int_{A_\Gamma} \dot{m} da \quad . \end{aligned}} \tag{2.30}$$

2.4 Avoiding the sharpness: the Level-set methods

The sharp transport of f is very challenging numerically (see Section 3.1 for a complete discussion). Therefore, it can be replaced by the transport of ϕ, which is smooth and regular in the vicinity of the interface. Then, the interface is located by the zero-isocontour of ϕ, and the transport of this quantity is not related to the transport of χ_l anymore. Moreover, it is not possible to replace χ_p directly by ϕ in Eq. (2.15) to correctly derive the two-phase equations as these quantities do not share the same properties.
To mimic the definition of χ_p without transporting it, a Heaviside function H_Γ can be reconstructed from ϕ. It is important to notice that H_Γ is not transported and is just an intermediate quantity used to supplant χ_l in Eq. (2.15):

$$\begin{aligned}\frac{\partial \rho H_\Gamma \Phi}{\partial t} + \nabla \cdot (\mathbf{u} \rho H_\Gamma \Phi) =& - \nabla \cdot H_\Gamma \mathbf{F} + \rho H_\Gamma S \\ &+ (\rho \Phi (\mathbf{u} - \mathbf{u}_\Gamma) + \mathbf{F}) \cdot \nabla H_\Gamma \quad . \end{aligned} \tag{2.31}$$

By adding equation Eq. (2.31) for both phases, the following equation is obtained:

$$\boxed{\begin{aligned}\rho^V \left(\frac{\partial \Phi^M}{\partial t} + \nabla \cdot \left(\mathbf{u}^M \Phi^M\right)\right) =& - \nabla \cdot \mathbf{F}^V + \rho^V H_\Gamma S^V \\ &+ \left([\rho \Phi (\mathbf{u} - \mathbf{u}_\Gamma)]_\Gamma + [\mathbf{F}]_\Gamma\right) \cdot \nabla H_\Gamma \quad . \end{aligned}} \tag{2.32}$$

Then Eq. (2.31) or Eq. (2.32) can be applied to the governing equations in Section 1.3 to obtain a one-fluid or two-fluid derivation. The averages defined with Eq. (2.23) are adapted to this new representation and the interface closures are also derived accordingly

$$\Phi_p^V = \Phi_g + H_\Gamma \left[\Phi\right]_\Gamma \quad , \tag{2.33a}$$

$$\Phi_p^M = \frac{\rho_g \Phi_g + H_\Gamma \left[\rho\Phi\right]_\Gamma}{\rho^V} \quad , \tag{2.33b}$$

$$\Phi_p^H = \frac{\rho_g c_{p,g} \Phi_g + H_\Gamma \left[\rho c_p \Phi\right]_\Gamma}{\rho^V c_p^M} \quad . \tag{2.33c}$$

Note that in order to obtain Eq. (2.31), a strong assumption has been made on the evolution of H_Γ which is considered to be transported with the velocity $\mathbf{u}_\Gamma$. This assumption is not exactly true as H_Γ is a quantity obtained from the transport of ϕ at velocity $\mathbf{u}_\Gamma$. While this choice leads to the loss of consistency between the interface transport and the averaged equations, it constitutes a real interest for the numerical resolution of the interface transport.
As H_Γ is not transported, it can take arbitrary forms as long as they are comprised between 0 and 1.

2.4.1 Sharp Heaviside formulation

In most modern Level-Set implementations of the governing two-phase flow equations, the Heaviside is taken as the sharpest possible on a computational grid. It takes the following form

$$H_\Gamma^0(\phi) = \begin{cases} 1 & \text{if } \phi > 0 \\ 0 & \text{otherwise} \end{cases} \quad . \tag{2.34}$$

With this choice of Heaviside, all the averages are equivalent and expressed as

$$\Phi^0 = \Phi^V = \Phi^M = \Phi^H = \begin{cases} \Phi_l & \text{if if } \phi > 0 \\ \Phi_g & \text{if otherwise} \end{cases} \quad . \tag{2.35}$$

Then this gives a system of equation where Φ_l, and Φ_g are solved separately in their respective domain, and the coupling at the interface is done by ensuring either a jump condition or a boundary condition. Then the discrete operators applied to Φ_l and Φ_g in the interface neighbourhood ensure explicitly the jump condition or boundary condition at the interface using Ghost Fluid Methods (GFM) or Ghost Cells Method (GCM) detailed in Appendix A.1 and Section 8.3.2, respectively.

2.4.2 Smooth Heaviside formulation

Another classical choice for the Heaviside function H_Γ^ϵ is based on the following requirement

$$\lim_{\epsilon \to 0} H_\Gamma^\epsilon \to \chi_l \tag{2.36}$$

In a discrete form, this means that if ϵ explicitly depends on the mesh size, then H_Γ^ϵ converges to χ_l with mesh resolution. This methodology has been employed in the early LS incompressible two-phase flow solvers [214] to improve accuracy in the presence of important density jumps, and is still used in LS solvers to treat the velocity jump for evaporation simulations [107].
For this purpose, a smoothed Heaviside computed from ϕ is often used

$$H_\Gamma^\epsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ 1 & \text{if } \phi > \epsilon \\ 1 + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & \text{otherwise} \end{cases}, \tag{2.37}$$

with ϵ an interface width often depending on the mesh size Δx. This formulation is very similar to the VOF one-fluid formulation presented in Section 2.3.2 although $H_\Gamma^\epsilon(\phi)$ is a consequence of huge simplifications in the derivations.

2.4.3 Consistent transport of the Heaviside

As stated at the beginning of this section, if the Heaviside H_Γ is not transported, no closure naturally arises from the two-fluid derivation and the choices are made from numerical considerations. However, it is possible to improve the consistency of the two-fluid and one-fluid formulations by transporting H_Γ. This has been done in the literature by using the hyperbolic tangent $H_\Gamma = \psi$ such that

$$\psi = \frac{1}{2}\left(\tanh\left(\frac{\phi}{2\epsilon}\right)\right) \tag{2.38}$$

with ϕ the distance function and ϵ the interface thickness. This form of Heaviside derives from two different methods:

- A method derived in a phase-field fashion with ψ used as a color function related to a measure of phase [88; 209].
- The conservative LS, with ψ an alternative level-set used to improve conservation of the interface mass [150; 38]

They both use ψ but with a different original meaning. In incompressible phase-field methods, ψ is transported with a source term derived from Cahn-Hilliard equations. However, it has been remodelled in [136] to the form:

$$S = \nabla \cdot \left(\mathcal{U}\left(\epsilon\nabla\psi - \psi\left(1-\psi\right)\frac{\nabla\psi}{|\nabla\psi|}\right)\right) \tag{2.39}$$

with $\mathcal{U}$ a free parameter depending on the fluid velocity.
Recently, an improvement has been proposed in [136] where this source term appears as an additional term in the one-fluid governing equations to be consistent with the transport of ψ. By deriving the correct one-fluid equations, consistency and strict conservation of mass, momentum and kinetic energy is retrieved in the case of two-phase incompressible flows without phase-change.

Another method proposed recently in [11] transports ψ and then apply a second step called reinitialization [48] to enforce the correct interface width ϵ

$$\frac{\partial \psi}{\partial \tau} + \nabla \cdot (\psi(1-\psi)\mathbf{n}_\Gamma) = \nabla \cdot (\epsilon \left(\nabla \psi \cdot \mathbf{n}_\Gamma\right) \mathbf{n}_\Gamma) \tag{2.40}$$

To improve consistency, a reinitialization step is also required for the governing equations. It appears as an additional evolution equation in the one-fluid and two-fluid equations compared to the original VOF formulations. This new formulation has been applied to the two-fluid energy equations to have consistent energy transfer at the interface and enforce energy conservation in the respective phases.
These methods constitute a significant improvement toward the proper derivation of two-phase equations for other representations than the sharp volume fraction. However, they do not directly derive from the indicator function as they do not result from an average procedure of Section 2.1.3 or from a discretization process such as finite-volume methods in Section 2.3.

2.5 Toward non-sharp physically-grounded equations

This last section aims to detail the pathway to derive a non-sharp derivation of two-fluid and one-fluid equations based on the averaging of the transport equation of χ_p and Eq. (2.15) presented in Section 2.1.3. This can be done by averaging Eq. (2.15) with the same average operator $\langle \cdot \rangle$ used to derive Section 2.1.3:

$$\begin{aligned}\frac{\partial \alpha_p \rho_p^i \Phi_p^m}{\partial t} + \nabla \cdot \left(\alpha_p \rho_p^i \mathbf{u}_p^m \Phi_p^m\right) = -\nabla \cdot \left(\alpha_p \mathbf{F}_p^i\right) + \alpha_p S_p^i \\ + \langle (\rho_p \Phi_p \left(\mathbf{u}_p - \mathbf{u}_\Gamma\right) + \mathbf{F}_p) \cdot \nabla \chi_p \rangle \quad .\end{aligned} \tag{2.41}$$

In these equations, two new averages are introduced as the intrinsic-averaged quantity Φ_p^i and mass-averaged quantity Φ_p^m of Φ associated to phase p:

$$\Phi_p^i = \frac{\langle \chi_p \Phi \rangle}{\alpha_p} \quad , \tag{2.42}$$

$$\Phi_p^m = \frac{\langle \chi_p \rho \Phi \rangle}{\alpha_p \rho_p^i} \quad , \tag{2.43}$$

Note that the present framework's density is constant in a given phase p and $\Phi_p^i = \Phi_p^m$. Moreover, as Φ_p^m, $\mathbf{u}_p^m$, $\mathbf{F}_p^i$ and S_p^i are always phase or mass-averaged in this process, the superscript m and i are dropped for clarity.
Then, the interface contribution can be rewritten as $\gamma_p(\Phi_p)$ which consists of a convective contribution from the mass transfer at the interface and a flux contribution related to Φ_p at the interface. Thus, the two-fluid instantaneous local balance is

$$\boxed{\frac{\partial \alpha_p \rho_p \Phi_p}{\partial t} + \nabla \cdot (\alpha_p \rho_p \mathbf{u}_p \Phi_p) = -\nabla \cdot (\alpha_p \mathbf{F}_p) + \alpha_p S_p + \dot{\gamma}_p \left(\Phi_p\right) \quad .} \tag{2.44}$$

In the derivation of two-fluid governing equations, the average $\langle\cdot\rangle$ have to be defined in order to close the interface contribution $\dot{\gamma}_p\left(\Phi_p\right)$. Moreover, Eq. (2.11) needs to be solved to evolve α_p which also contains contributions at the interface depending on the average $\langle\cdot\rangle$.
By defining the correct average procedure $\langle\cdot\rangle_\psi$ allowing to go from χ_l to ψ, then the additional terms arising from this averaging would appear in $\dot{\gamma}_p\left(\Phi_p\right)$ and Eq. (2.11) leading to a proper derivation of the two-phase equations. Unless this strict relation is established, the direct replacement of χ_l by ψ in the equations neglects several interface contributions due to the exchange of quantities at the interface. The same ideas can be advanced for algebraic VOF.
This is a very challenging mathematical formulation for which no further details are provided in this manuscript. However, it represents a promising lead to developing of more consistent two-phase flow formulations prior to numerical discretization.

2.6 Definition of the formulations used in this work

The different frameworks presented in this chapter compose the basis of all the numerical methods used in this work. More specifically, the formulations are divided into three paradigms:

1. The Whole Domain Formulation (WDF) where the transported quantities are naturally averaged with the general operators Eq. (2.33). Then the interface contributions are embedded in source terms only located in the vicinity of the interface.
2. The Jump Condition Formulation (JCF) where the transported quantities are explicitly separated by choosing a sharp representation of the Heaviside in Eq. (2.33). Then, the closures at the interface are performed using specific numerical methods to impose jump conditions or boundary conditions.
3. The Two-fluid Formulation (TFF) which directly derives from the finite-volume discretization of the equations. The quantities of both liquid and gas phases are then simultaneously transported and available at the interface.

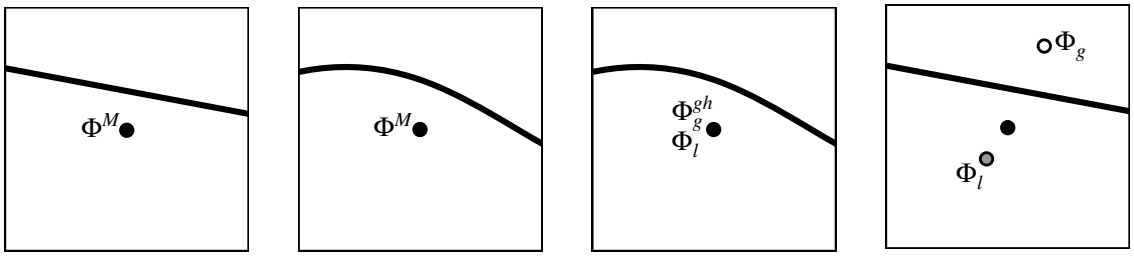
The choice has been made to include the VOF one-fluid formulation derived in Section 2.3.2 in the WDF for simplicity. It is crucial to keep in mind that WDF is notional and only means that the quantities transported using this formulation are averaged either by Eq. (2.23c) with different definitions of H_Γ^ϵ. In the case of LS, H_Γ^ϵ will be defined with Eq. (2.37) while in VOF it will be taken as f.
Another notation used for convenience to bridge all the formulations in the remaining of the manuscript is the notational relation

$$\frac{1}{Vc}\int_{A_\Gamma} da = \nabla H_\Gamma \cdot \mathbf{n}_\Gamma = \delta_\Gamma \quad , \tag{2.45}$$

which can be applied to WDF, JCF or TTF.

2.6.1 Clarifications on WDF, JCF and TFF

The illustration in Section 2.6.1 shows the difference between the three formulations applied to a finite-volume discretization.



(a) *WDF with VOF* **(b)** *WDF with LS* **(c)** *JCF* **(d)** *TFF*

Figure 2.1: *Illustration of the different formulations for a quantity Φ in a control volume. The black point is the center of the cell while the grey point is the liquid phase barycenter and the white point is the gas barycenter.*

While WDF only provides the information of an averaged quantity at the interface (here, the mass average is used, but it could be volume or enthaply average depending on the considered equation), JCF provides the information of one phase only (and needs to reconstruct the other one) and the TFF provides the quantity information in the two phases. Moreover, WDF has been represented for both VOF in Fig. C.1a and LS in Fig. C.1b to emphasise that they are not equivalent as VOF also contains the information of the volume fraction in the cell while the LS does not.

2.6.2 Formulations applied to the governing equations

It is also important to remind that the formulations can be independently applied to any of the governing equations in the system.
More specifically, the mass and momentum equations are always treated with WDF or JCF in this work (and in almost every low Mach derivations of the literature). This choice emerges from the representation of the pressure in the majority of the low Mach solvers. As the pressure is not related to a thermodynamic quantity, it is often treated as a quantity of arbitrary magnitude which only have a meaning for the gradient ∇P (this point is detailed in Section 5.1). Then, the explicit value of P cannot be used in the equations because it only holds a numerical meaning. This is not an issue for WDF and JCF, as the interface contribution related to pressure is hidden in the surface tension term of Eq. (2.26).
On the other hand, if momentum had to be treated with a TFF, then the value P appears in the interface contributions and requires a correct definition. The evaluation of such a value of pressure needs further closures which have to be determined.
In the case of energy or mass species fraction, WDF, JCF, or TFF can be applied without distinction as every quantity appearing in the interface contributions can be closed from the set of governing equations.

In the remainder of the manuscript, a formulation will always refers to a specific equation and not to the entire set of governing equations.

2.7 Conclusion

In this chapter, the governing equations have been derived for two-phase flows applying the finite-volume method on the two-fluid equations. This derivation provides a natural equation set in a geometric VOF framework where all interface contributions are closed using the volume fraction f. In the LS framework, further assumptions need to be made to obtain the governing equations of two-phase flows. However, recent advances have been made in the derivation of the LS two-phase equations by transporting H_Γ. This has been done a source term or a two-step evolution of the equation, which need to be integrated into the set of governing equations to keep the consistency of the system. Then, the general derivation of the two-fluid equations has been provided to start a discussion on how to proceed towards non-sharp physically-grounded equations for two-phase flows. Finally, these formulations are presented in this specific work based on a finite-volume discretization.
Now that the mathematical formulation is complete, the following parts detail the numerical considerations concerning the resolution of the governing equations with finite-volume methods on cartesian grid.

Part II

Numerical representation of a sharp interface

Chapter 3

Interface capturing methods

Contents

The first part of this work considers the transport of the interface without considering the coupling with the flow. The VOF and LS interface capturing methods evoked in Chapter 2 solve Eq. (2.2) with a discrete representation of χ_l, referred to as the color function c. Regardless of the interface capturing method choice, it results in solving an advection equation for c

$$\frac{\partial c}{\partial t} + \mathbf{u}_\Gamma \cdot \nabla c = 0 \quad . \tag{3.1}$$

Note that for geometric VOF, this is a purely notional form used for clarity.
In the context of a divergence-free flow without phase-change, interface velocity is equivalent to the flow velocity. Eq. (3.1) can thus be rewritten as a conservation equation of c

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma c) = 0 \quad . \tag{3.2}$$

The color function can be defined in various ways ,which all require specific numerical treatments. In the following, details are given about flux computation and eventual additional steps in the advection process. The goal of this study is to select promising interface capturing methods from the literature for the simulation of two-phase flows. The potential candidates are chosen using the following criteria:

- Up-to-date or still relevant in recent solvers;
- Adapted to a finite volume cartesian grid framework;
- Reasonable implementation effort;

In Section 3.1 and Section 3.2 are presented the VOF and Standard LS (SLS) methods respectively. Then, two alternative methods are also described: another class of LS method, the conservative LS (CLS) is detailed in Section 3.3 and finally, the coupling between LS and VOF (CLSVOF) is introduced in Section 3.4 for completeness.

3.1 Volume-Of-Fluid method

The VOF method directly arises from the finite-volume method derivation presented in Section 2.1.1. c is then the volume average of χ in a computational cell $\mathcal{C}$

$$f = \frac{\int_{\mathcal{C}} \chi_l dv}{V_{\mathcal{C}}} \quad , \tag{3.3}$$

where $V_{\mathcal{C}}$ is the volume of $\mathcal{C}$.
The volume fraction is governed by Eq. (2.7) which is rewritten such that $\alpha_p = f$ and phase-change does not occur:

$$\frac{\partial f}{\partial t} + \frac{1}{V_{\mathcal{C}}} \int_{\partial V_{\mathcal{C}}} \chi_l \mathbf{u} \cdot \mathbf{n} da = 0 \quad . \tag{3.4}$$

In the following, the notional formulation of Eq. (3.2) with $c = f$ is used instead for clarity

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma f) = 0 \quad . \tag{3.5}$$

Two main approaches were developed to solve Eq. (3.5):

- Algebraic VOF [231] considers the resolution Eq. (3.5) using standard numerical fluxes either from a finite-element, finite-difference or finite-volume framework. The transport of volume fraction f requires flux limitation to remain bounded between 0 and 1 and suffers from numerical diffusion or dispersion. Major improvements have been made in the past decade to improve the algebraic VOF algorithms, for instance, with the multidimensional THINC method in both structured [84] and unstructured mesh [232]. Other works based on diffuse interface were also improved by the use of sharpening methods (see [200; 37] for details), which allow control on the interface thickness.

- Geometric VOF approach was first introduced by Hirt and Nichols [83]. The method was developed to prevail the limitations of the Marker and Cell (MAC) method developed by Harlow and Welch [77] in 1965 in which velocity field transports marker particles. The position of the interface is then deduced from the distribution of these markers in the domain: a cell containing markers with empty neighbouring cells is an interface cell. Moreover, the position of the marker particles gives a geometric description of the interface in the cell. This method was found to use a lot of storage resources and unnecessary huge computational time to move every single particle at each timestep. As an alternative, Hirt and Nichols proposed a new approach in which a color function is updated in the cell. The interface is reconstructed as a line and the normal to the interface is deduced from the color function gradient. This method requires only one variable per cell to fully describe the interface propagation, which is a massive gain in computational time and storage.

Despite their increasing interest in two-phase flow simulations, the algebraic VOF methods are incompatible with a sharp representation of the interface in the sense of the strict derivation of Section 2.3. Therefore, our study focuses on geometric VOF, which requires a reconstruction strategy to evaluate the fluxes of χ_l in Eq. (3.4). The methods used to reconstruct the interface in the mixed cells ($0 < f < 1$) are presented in the next section.

3.1.1 Reconstruction

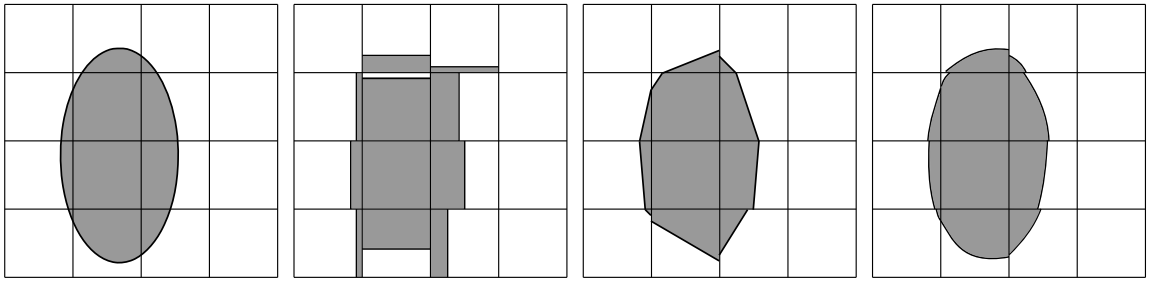
The most basic reconstruction is a line aligned with the mesh grid as in Fig. 3.1b. This method is known as the Simple Line Interface Construction (SLIC) and was introduced by Noh and Woodward [148]. Even if such a representation is first order, it has been widely used in early two-phase flow solvers and has given reasonable results for a small cost. Note that in this technique, the interface is not explicitly reconstructed. The line direction is defined in the cell while computing the fluxes and then alternately modified at each timestep in order to minimize the first-order error.
In modern codes, higher-order reconstructions are commonly used. The most popular is the Piecewise Linear Interface Calculation (PLIC) which allows the line to have an arbitrary direction as in Fig. 3.1c. For PLIC reconstruction, the following plane equation is considered:

$$\mathbf{n} \cdot \mathbf{x} = d \quad , \tag{3.6}$$

with $\mathbf{n}$ the normal to the interface pointing out of the dark fluid and d the line parameter. In the following, details are provided on the strategies to obtain $\mathbf{n}$ and d from the VOF representation.

3.1.1.1 Normal evaluation

In SLIC, Eq. (3.6) reduces to $x = 0$, $y = 0$ or $z = 0$ while a PLIC reconstruction requires to compute $\mathbf{n}$ from f. PLIC reconstructions have an accuracy between first and second-order. Second-order reconstruction is achieved if the algorithm represents any line (or plane in 3D) exactly [181]. More details can be found in the Pilliod and Puckett's review of reconstruction algorithms [168].



(a) *Exact* **(b)** *SLIC* **(c)** *PLIC* **(d)** *PROST*

Figure 3.1: *Different reconstruction methods for an ellipse on a cartesian mesh. The grey part represent the liquid phase while the white part is the gas phase.*

Parker and Youngs The most straightforward computation is the Parker and Youngs method [161] which uses the gradient of f to evaluate the normal components. In 2D, a 3x3 block is considered and f is interpolated at the 4 vertices, for the vertex $\left(i+\frac{1}{2}, j+\frac{1}{2}\right)$ the normal components are

$$n_x = -\frac{\overline{f}_{i+1,j+\frac{1}{2}} - \overline{f}_{i,j+\frac{1}{2}}}{\Delta x} \quad ,$$
$$n_y = -\frac{\overline{f}_{i+\frac{1}{2},j+1} - \overline{f}_{i+\frac{1}{2},j}}{\Delta y} \quad .$$

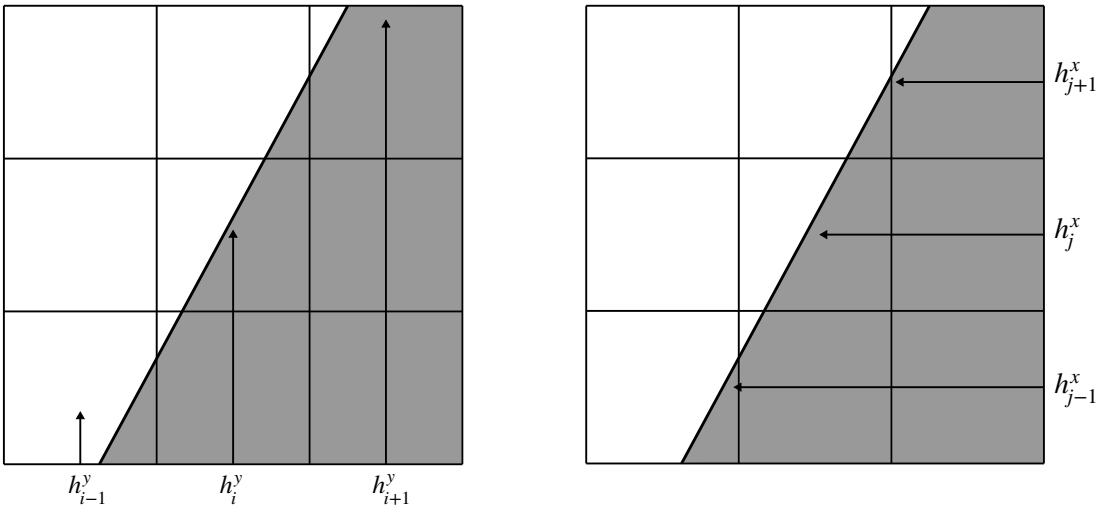
$\overline{f}_{i,j+\frac{1}{2}}$ is the vertex interpolation of f such that $\overline{f}_{i,j+\frac{1}{2}} = \frac{1}{2}\left(f_{i,j} + f_{i,j+1}\right)$.
The normal is finally evaluated by averaging the 4 vertex normals:

$$\mathbf{n} = \frac{1}{4}\left(\mathbf{n}_{i-\frac{1}{2},j-\frac{1}{2}} + \mathbf{n}_{i+\frac{1}{2},j-\frac{1}{2}} + \mathbf{n}_{i-\frac{1}{2},j+\frac{1}{2}} + \mathbf{n}_{i+\frac{1}{2},j+\frac{1}{2}}\right) \quad . \tag{3.7}$$

This normal computation is interesting for its simplicity and its straightforward extension to 3D. Unfortunately, it does not achieve second-order accuracy.

Centered Columns For this method, the normal is computed from the centered difference of the height function over a direction in a 3x3 block. Height function is a crucial notion first introduced in the SOLA-VOF program to treat complex topology with VOF [146] and is widely used for normal and curvature computations (see [19] for details). Height function can be computed from various stencil, and here the method is based on a compact 3-stencil height function computed in all directions. It is the sum of the volume fractions over a column as represented on Fig. 3.2

$$h_j^x = \sum_{k=-1}^{1} f_{i+k,j} \quad , \qquad h_i^y = \sum_{k=-1}^{1} f_{i,j+k} \quad , \tag{3.8}$$



(a) *Height function in the y direction* **(b)** *Height function in the x direction*

Figure 3.2: *Height function in a* 3×3 *compact block. It is either reconstrcuted in the vertical direction (left) or the horizontal direction (right).*

For the y direction, CC method gives

$$n_x^{CC,y} = \frac{1}{2}\left(h_{i+1}^y - h_{i-1}^y\right) \quad ,$$
$$n_y^{CC,y} = sign(-\left(f_{i,j+1} - f_{i,j-1}\right)) \quad .$$

To select the best normal, the normal components are normalized such that $|n_x| + |n_y| = 1$ and use the following criterion.

$$\max\left(\left|n_x^{CC,x}\right|, \left|n_y^{CC,y}\right|\right) \quad . \tag{3.9}$$

This approach remains relatively easy to implement and improve the accuracy of the normal evaluation compared to PY. However, it still does not achieve second-order accuracy.

MYC The Mixed Youngs Centered is a good compromise between accuracy and efficiency [135]. A first normal $\mathbf{n}^{PY}$ is evaluated with the Parker and Youngs method, and the second normal $\mathbf{n}^{CC}$ is evaluated with the Center Columns method described previously. Then, the normal that minimizes the component of the leading direction of the kept centered column normal is chosen. For example, if $\mathbf{n}^{CC} = \mathbf{n}^{CC,x}$ then the criterion is:

$$\min\left(n_x^{CC}, n_x^{PY}\right) \quad . \tag{3.10}$$

Note that all the normal components are again normalized such that $|n_x| + |n_y| + |n_z| = 1$ when applying the criterion. This is a further improvement of PY and CC methods which improve the accuracy without reaching second-order reconstructions.

WY Based on simple observations of the behaviour of height function regarding the cell center value of the volume fraction f_c, Weymouth and Yue introduced a second-order version of the CC method. The following observations are given

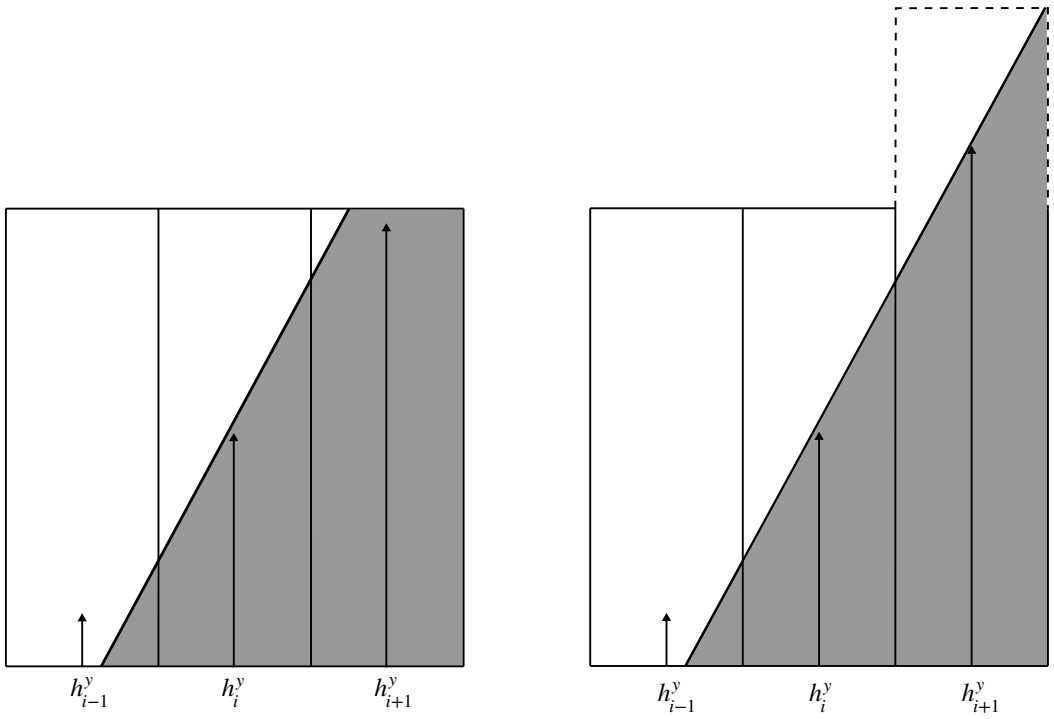
- The middle height function of a 3×3 block is always estimated with second-order accuracy.
- if $f_c > 0.5$ the left height function is also estimated with second-order accuracy while the right height function is under-estimated
- if $f_c < 0.5$ the right height function is second-order accurate while the left one is under-estimated.

The normal is then estimated using either forward or backward difference depending on f_c. In the case of Fig. 3.3, $f_c > 0.5$ and using a forward difference allows to reconstruct exactly the line. This method has the ability to reconstruct all lines/planes exactly and then achieves second-order accuracy for all possible configurations.

ELVIRA Finally, another second-order reconstruction was derived by Pilliod in [167]. The Efficient Least squares VOF Interface Reconstruction Algorithm is an extension of the LVIRA method, first introduced by Puckett in [174]. The LVIRA was an iterative method whereas the efficient version reduces the cost of the method by minimizing the error over only 6 (27 in 3D) normal candidates. The candidates are the forward, centered and backward differences of the height value in both directions

$$
\begin{aligned}
n_x^{f,x} &= \sum_{l=-1}^{1} (f_{i+1,j+l} - f_{i,j+l}) & n_y^{f,x} &= 1 \quad , \\
n_x^{c,x} &= \frac{1}{2} \sum_{l=-1}^{1} (f_{i+1,j+l} - f_{i-1,j+l}) & n_y^{c,x} &= 1 \quad , \\
n_x^{b,x} &= \sum_{l=-1}^{1} (f_{i,j+l} - f_{i-1,j+l}) & n_y^{b,x} &= 1 \quad , \\
n_y^{f,y} &= \sum_{k=-1}^{1} (f_{i+k,j+1} - f_{i+k,j}) & n_x^{f,y} &= 1 \quad , \\
n_y^{c,y} &= \frac{1}{2} \sum_{k=-1}^{1} (f_{i+k,j+1} - f_{i+k,j-1}) & n_x^{c,y} &= 1 \quad , \\
n_y^{b,y} &= \sum_{k=-1}^{1} (f_{i+k,j} - f_{i+k,j-1}) & n_x^{b,y} &= 1 \quad .
\end{aligned}
$$

It can be shown that the ELVIRA algorithm reconstructs all linear interfaces exactly, and it is then second-order accurate. This version of the algorithm is commonly used [156]. Unfortunately, a block of $5 \times 5 \times 5$ cells is required to preserve the second-order accuracy in 3D with 72 to 144 normal candidates [133].
Other algorithms have been introduced more recently to keep a compact block of $3 \times 3 \times 3$ such as CVTNA [115], or a reduced version of ELVIRA in a $3 \times 3 \times 3$ block [28].



(a) *Height function on the compact* 3×3 *block* **(b)** *Exact height function values*

Figure 3.3: *Height function error when* $f_c > 0.5$*. The left configuration under-estimates the height function* h^y_{i+1} *with the missing part represented in dashed line on the right.*

Moment of fluid Moment of fluid (MOF) method has been first introduced by Shashkov et al. [55] in an unstructured mesh framework. It is an extension of VOF where the center of mass of the fluid $\mathbf{x}_{cm}$ is also used for interface reconstruction. The center of mass is a new sub-cell information which allows compact and accurate interface normal computation. This is of huge interest in under-resolved cases.
One can show that the f and $\mathbf{x}_{cm}$ are directly related to the lower moments of the indicator function χ_l. Suppose that we have fluid in a cell $\mathcal{C}$, one can define the zeroth moment of the fluid as

$$m_0(\chi_l) = \int_{\mathcal{C}} \chi_l(\mathbf{x}) dv \quad . \tag{3.11}$$

Then the volume fraction f is directly deduced as $f = \frac{m_0(\chi_l)}{V_{\mathcal{C}}}$.
Now, the first-order moments are defined as

$$\mathbf{m}_1(\chi_l) = \int_{\Omega} \mathbf{x}\chi_l(\mathbf{x}) dv \quad . \tag{3.12}$$

Hence, the center of mass of the fluid is $\mathbf{x}_{cm} = \frac{\mathbf{m}_1(\chi)}{m_0(\chi)}$.
In MOF, the reference center of mass $\mathbf{x}^{ref}_{cm}$ is transported with the volume fraction. The

actual center of mass $\mathbf{x}_{cm}^{act}$ is the one obtained by the reconstruction that ensures mass conservation, it is a function of d and $\mathbf{n}$ giving the error function to minimize:

$$E(d, \mathbf{n}) = \left\| \mathbf{x}_{cm}^{act}(d, \mathbf{n}) - \mathbf{x}_{cm}^{ref} \right\|_2 \quad . \tag{3.13}$$

The parametric notation of $\mathbf{n}$ is considered

$$\mathbf{n} = \begin{pmatrix} \cos(\Theta_1)\sin(\Theta_2) \\ \sin(\Theta_1)\sin(\Theta_2) \\ \cos(\Theta_2) \end{pmatrix} \quad , \tag{3.14}$$

with Θ_1 and Θ_2 the azimuthal and polar angles respectively.
The minimization can be done using the Gauss-Newton method as in [140] which only uses first-order derivative of the objective functions (no improvement was found by using the Newton minimization using second-order derivatives). This can also be done by using Hailey's method [35] or Broyden-Fletcher-Goldfarb-Shanno [237]. This development has also been done in a slightly different way by Lemoine et al. [108] with an analytical approach.

3.1.1.2 Plane parameter d

Once normal components are computed, the remaining unknown is the line parameter d. It has been shown in [193] that an analytic expression for d exists such that the f-constraint is enforced by using a cube-chopping algorithm. This process is able to give d or f in all possible configurations analytically. It is a noticeable observation that avoids the use of an iterative method leading to more computation time and truncation errors. Note that this problem can also be solved with an iterative method [181].

3.1.1.3 Conclusion on reconstruction

Even if the PLIC reconstruction is widely used in two-phase flow simulations, it is limited to second-order accuracy. Some works have been directed about parabolic reconstruction such as PROST method of Renardy and Renardy [178] (see Fig. 3.1d) or QUASI [50]. The first motivation of such a method is to have an inherent curvature in the reconstruction in opposition to PLIC. It improves the accuracy of advection fluxes but requires significant additional implementation efforts and computational cost. Extension to 3D is also not straightforward and has not yet been presented in the literature.
From the above considerations, the most promising reconstructions are the ELVIRA and MOF methods which provide a second-order evaluation of $\mathbf{n}$. However, MOF does not seem to provide substantial improvements compared to VOF-based methods for our target applications. Indeed, this method seems to improve robustness in the case of atomization, where most of the droplets are under-resolved. An appropriate choice is to use ELVIRA for $\mathbf{n}$ evaluation while d is deduced analytically using the cube-chopping relations of [193] to avoid additional computational cost and threshold errors of an iterative process.

3.1.2 Flux computation

Classical advection schemes suffer from numerical diffusion or dispersion, which leads to the smearing of the interface or overshoot and undershoot of the volume fraction. In both cases, the sharpness and consistency of the interface is lost.
To avoid this problem, the transport of the interface is evaluated with geometrical methods. There are two types of geometric VOF advection schemes. The first is known as split method, where the fluxes are computed in a direction-by-direction fashion. They are limited to cartesian grids and suffer from additional numerical errors due to the dimensional splitting of the advection process. However, they keep a simple implementation and can achieve second-order accuracy using a Strang splitting [206]. Even if the 3D extension seems straightforward, keeping exact mass conservation is challenging using split method as discussed in [13]. On the other hand, the unsplit type methods are more accurate, but their implementation can become really complex involving simplex decomposition. These methods are even more complex when dealing with 3D extension, and also need additional treatments to ensure exact mass conservation in complex flows (see [118], [156] for details). Accordingly, the choice has been made to use a split scheme in the present work to limit the computational cost of the interface capturing. The dimensional-splitting of Eq. (3.5) results in solving successive 1D advection problems

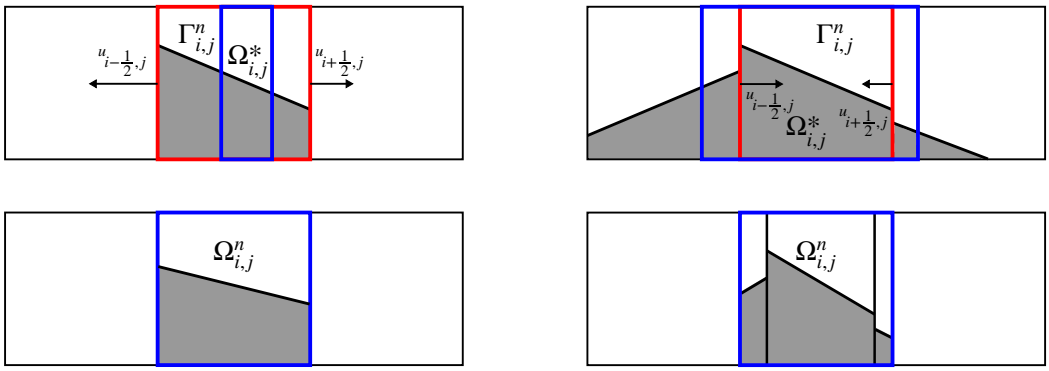
$$\frac{\partial f}{\partial t} + \frac{\partial \left(u_s f\right)}{\partial x_s} = f\frac{\partial u_s}{\partial x_s} \quad , \tag{3.15}$$

with s the sweep direction. $\nabla \cdot \mathbf{u} = 0$ does not imply $\frac{\partial u_s}{\partial x_s} = 0$ for a given splitting direction and has to appear explicitly in the RHS of Eq. (3.15). After each directional splitting step, a reconstruction is performed in order to avoid double fluxing problems [181].
The geometrical flux approximation can take different forms. Historically, Eulerian and Lagrangian schemes were developed with comparable accuracy. For a directional-splitting method, the challenge of geometrical flux approximation does not reside on the accuracy but on the conservation and boundness properties. The following section describes in detail the complexity to achieve conservation in such a framework.

3.1.2.1 EI-LE method

The Eulerian Implicit-Lagrangian Explicit method was first presented by Aulisa et al. [13]. It uses a combination of Eulerian and Lagrangian schemes leading to exact mass conservation in 2D. However, extension in 3D is not straightforward.

Eulerian implicit scheme The EI scheme was first introduced by Rider and Kothe in [181]. The initial computational cell $\Omega^n_{i,j}$ is mapped to $\Omega^*_{i,j}$ (blue box in Fig. 3.4) and all the interfaces lying into this area are then resized to the initial computational cell $\Omega^n_{i,j}$.
The $\Omega^*_{i,j}$ limits are $x \in \left[x_{i-\frac{1}{2},j} - u_{i-\frac{1}{2},j}\Delta t, x_{i+\frac{1}{2},j} - u_{i+\frac{1}{2},j}\Delta t\right]$ and $y \in \left[y_{i,j-\frac{1}{2}}, y_{i,j+\frac{1}{2}}\right]$.



(a) *Dilatation case* **(b)** *Compression case*

Figure 3.4: *Illustration of EI flux computation. The computational cell $\Omega_{i,j}$ is represented in blue while the control volume related to the interface $\Gamma_{i,j}$ is in red.*

$f^*_{i,j}$ is then computed using

$$f^*_{i,j} = \frac{\Omega^*_{i,j} \cap \Gamma^n_{i-1,j} + \Omega^*_{i,j} \cap \Gamma^n_{i,j} + \Omega^*_{i,j} \cap \Gamma^n_{i+1,j}}{\left(1 - \frac{\Delta t}{\Delta x}\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)\right)\Delta x \Delta y} \quad . \tag{3.16}$$

In Fig. 3.4, one can see that for the dilatation case only $\Omega^*_{i,j} \cap \Gamma^n_{i,j}$ is non-zero while for the compression case all three intersections are non-zero. All the areas used in Eq. (3.16) can be easily retrieved using the analytic expression between f and d given in [193].

Lagrangian explicit scheme The LE scheme is due to Li [113]. Velocities are considered constant at the faces of the cell, and the interface is propagated to the neighbouring cells. This scheme takes implicitly into account $\frac{\partial u}{\partial x}$ when it is not zero. In this special case, the interface propagation is not a simple translation but also a compression or dilatation of the interface resulting in a modified line equation with new normal components [73]. The explicit mapping gives

$$x^n = \frac{x^* - u_{i-\frac{1}{2},j}\Delta t}{1 + \frac{\Delta t}{\Delta x}\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)} \quad , \tag{3.17}$$

with x^n a point in cell $\Omega^n_{i,j}$ at time n and x^* the same point after the x-step. By introducing Eq. (3.17) in the line equation

$$n^n_x \left(\frac{x^* - u_{i,j}\Delta t}{1 + \frac{\Delta t}{\Delta x}\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)}\right) + n^n_y y = d^n \quad , \tag{3.18}$$

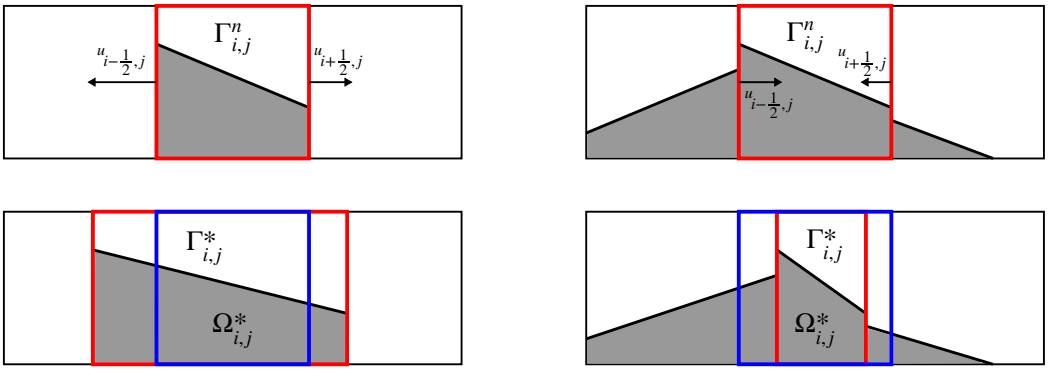
resulting in a new line equation

$$n^*_x x^* + n^*_y y^* = d^* \quad , \tag{3.19}$$

with the new line parameters

$$n_x^* = \frac{n_x^n}{1 + \frac{\Delta t}{\Delta x}\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)} \quad , \tag{3.20}$$

$$d^* = d^n + n_x^* u_{i-\frac{1}{2},j}\Delta t \quad . \tag{3.21}$$



(a) *Dilatation case* **(b)** *Compression case*

Figure 3.5: *Illustration of LE flux computation. The computational cell $\Omega_{i,j}$ is represented in blue while the control volume related to the interface $\Gamma_{i,j}$ is in red.*

$f^*_{i,j}$ is computed using

$$f^*_{i,j} = \frac{\Omega^*_{i,j} \cap \Gamma^*_{i-1,j} + \Omega^*_{i,j} \cap \Gamma^*_{i,j} + \Omega^*_{i,j} \cap \Gamma^*_{i+1,j}}{\Delta x \Delta y} \quad . \tag{3.22}$$

In Fig. 3.5, one can see that for the dilatation case only $\Omega^*_{i,j} \cap \Gamma^*_{i,j}$ is non-zero while for the compression case all three intersections are non-zero. Again, analytical relations given in [193] are used to compute the different intersections.

Linear mapping interpretation The two schemes presented above do not ensure mass conservation when 1D dilatation or compression occurs [194]. This can be understood by taking these two schemes as pure linear mapping. As described in [17], it is possible to go back from the equation of motion $\frac{dx}{dt} = u(x(t))$. A first-order temporal integration gives $x^{n+1} = x^n + u(\widetilde{x})\Delta t$ with $\widetilde{x} = x^n$ in the explicit scheme and $\widetilde{x} = x^{n+1}$ in the implicit scheme. $u(\widetilde{x})$ is interpolated linearly between the face velocities $u_{i-\frac{1}{2},j}$ and $u_{i+\frac{1}{2},j}$:

$$u(\widetilde{x}) = (1 - \widetilde{x})u_{i-\frac{1}{2},j} + \widetilde{x}u_{i+\frac{1}{2},j} \quad . \tag{3.23}$$

This leads to two linear mappings Π_x^E and Π_x^I for the explicit and implicit schemes respectively

$$\Pi_x^E : \begin{cases} x_{i,j}^* = c_x^E x_{i,j}^n + u_{i-\frac{1}{2},j}\Delta t \\ y_{i,j}^* = y_{i,j}^n \end{cases} , \tag{3.24}$$

$$\Pi_x^I : \begin{cases} x_{i,j}^* = c_x^I x_{i,j}^n + c_x^I u_{i-\frac{1}{2},j}\Delta t \\ y_{i,j}^* = y_{i,j}^n \end{cases} , \tag{3.25}$$

with $c_x^E = 1 + \Delta t/\Delta x \left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)$ and $c_x^I = 1/1 - \Delta t/\Delta x \left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)$.
The linear mapping Π_x^E transforms the interface control volume $\Gamma_{i,j}^n$ into the rectangle $\Gamma_{i,j}^*$ in Fig. 3.5 while the linear mapping Π_x^I transforms the rectangular computational cell $\Omega_{i,j}^*$ into the square $\Omega_{i,j}^n$ in Fig. 3.4. Note that for 1D uniform flow field both mappings are equivalent as $c_x^E = c_x^I$.
For the EI-LE scheme, the EI step in x-direction can be written as

$$f_{i,j}^* = c_x^I \left(f_{i,j}^n - \left(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}\right)\right) \quad , \tag{3.26}$$

followed by the LE step in y-direction

$$f_{i,j}^{n+1} = c_y^E f_{i,j}^* - \left(F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}\right) \quad . \tag{3.27}$$

One combines Eqs. (3.26) and (3.27)

$$f_{i,j}^{n+1} = c_y^E c_x^I f_{i,j}^n - c_y^E c_x^I \left(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}\right) - \left(F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}\right) \quad . \tag{3.28}$$

By using the incompressibility condition $v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}} = -\left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)$:

$$c_y^E c_x^I = \frac{1 + \Delta t \left(v_{i,j+\frac{1}{2}} - v_{i,j-\frac{1}{2}}\right)}{1 - \Delta t \left(u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}\right)} = 1 \quad . \tag{3.29}$$

The update of f from Eq. (3.28) is then

$$f_{i,j}^{n+1} = f_{i,j}^n - \left(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}\right) - \left(F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}\right) \quad . \tag{3.30}$$

This last formulation is strictly conservative.
For the LE-EI scheme, the update is written as

$$f_{i,j}^{n+1} = f_{i,j}^n - c_y^I \left(F_{i+\frac{1}{2},j} - F_{i-\frac{1}{2},j}\right) - c_y^I \left(F_{i,j+\frac{1}{2}} - F_{i,j-\frac{1}{2}}\right) \quad , \tag{3.31}$$

which is not mass conservative for all c_y^I values.
Such a relation can also been derived for EI-EI or LE-LE schemes leading to the same conservation issues.

Mass loss for the different schemes To illustrate the mass conservation errors resulting from wrong combination of EI and LE steps, four different schemes have been tested: LE-LE, EI-EI, LE-EI and finally EI-LE.
Table 3.1 shows the mass error of these schemes for a 2D vortex-in-a-box test case (see Section 4.3.2 for details) on a $[64 \times 64]$ mesh. In fact, all schemes but EI-LE introduce mass errors even if the LE-EI approach is reducing drastically mass errors compared to LE-LE or EI-EI schemes.

Advection scheme	mass loss [%]
LE-LE	3.56
EI-EI	3.75
LE-EI	3.60e-6
EI-LE	4.71e-15

Table 3.1: *Mass loss for different split VOF schemes*

Unfortunately, there is no 3D combination providing exact mass conservation using EI and LE steps. Indeed, several combinations have been tested in [229] such as LE-LE-LE or EI-EI-LE, but all of them lead to mass error up to 1%. An attempt to a 3D extension was done in [12] by decomposing the 3D incompressible flow into 2D incompressible flows, but this technique requires to inverse a $(6N_x) \times (6N_y)$ matrix and cannot be applied to arbitrary solenoidal flows.

3.1.2.2 Weymouth and Yue method

More recently, a scheme has been proposed by Weymouth and Yue [228] (WY) where exact mass conservation is reached and 3D extension is straightforward. The method rely on EI fluxes applied to a modified form of the 1D advection Eq. (3.15)

$$\frac{\partial f}{\partial t} + \frac{\partial \left(u_s f\right)}{\partial x_s} = f_c \frac{\partial u_s}{\partial x_s} \quad . \tag{3.32}$$

In previous works, the compression/dilatation factor was taken as $f_c = f$ while this methodology consider f_c as

$$f_c = \chi(\mathbf{x}_c) = \begin{cases} 1 & \text{for } f > 0.5 \\ 0 & \text{else} \end{cases} \quad , \tag{3.33}$$

with f the volume fraction of liquid in the cell at the previous time step f^n (and not the previous directional splitting step !). f_c is then computed once at the beginning of the time step and remains constant until the next time step.
It can be shown that there are no undershoot or overshoot in any step of the directional splitting method with a Courant restriction of the form [229]:

$$\Delta t \sum_{i=1}^{N_{dim}} \left| \frac{u_i}{\Delta x_i} \right| < \frac{1}{2} \quad . \tag{3.34}$$

This ensures exact mass conservation of the transport for any velocity fields.
Note that another intent to provide exactly conservative method using directional splitting has been made in a redistribution-based fashion where the undershoots and overshoots are iteratively redistributed to the neighbouring cells [188]. This method was mainly developed to prevail the CFL restriction of WY and gives encouraging results on highly stretching test cases.
From the above considerations, the use of WY scheme seems to provide a robust method for VOF advection.

3.2 Standard Level-Set method

This method was first presented by Osher and Sethian [153] where ϕ is a signed distance transported in the whole domain, and the interface is located at the iso-contour $\phi^{-1}(0)$. Later, it was applied to the simulation of incompressible two-phase flows by Sussman et al. [214] as an alternative to the already existing VOF and FT algorithm. The main advantages of this formulation are its straightforward implementation in 2D and 3D, its smoothness allowing high order treatments and the natural handling of topology changes. The transport equation for the Level-Set is Eq. (3.2) with $c = \phi$

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma \phi) = 0 \quad . \tag{3.35}$$

3.2.1 Redistanciation

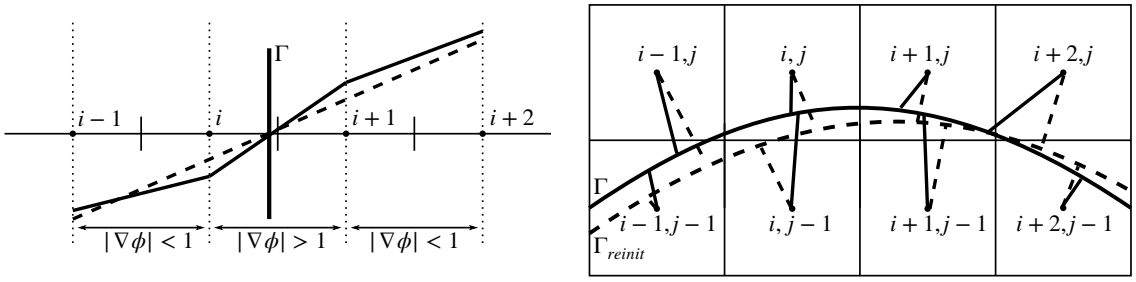
In the general case, $\mathbf{u}_\Gamma$ is not uniform in the normal direction of the interface. This causes severe deviations of ϕ from a signed distance leading to a decrease of the interface location accuracy [214]. After evolving ϕ with Eq. (3.35), additional numerical treatments are required to impose

$$|\nabla \phi| = 1 \quad . \tag{3.36}$$

One way to do that is to enforce Eq. (3.36) solving a second equation for ϕ. The following equation was first introduced in [214]

$$\frac{\partial \phi}{\partial \tau} + \text{sign}(\phi_0)\,(|\nabla \phi| - 1) = 0 \quad , \tag{3.37}$$

with τ a pseudo-time.
This results in solving a Hamilton-Jacobi equation in pseudo-time with the corresponding Hamiltonian $\mathcal{H}(\phi, \nabla\phi) = \text{sign}(\phi_0)\,(1 - |\nabla\phi|)$. Ideally, the redistanciation step would retrieve the constraint Eq. (3.36) without moving the interface Γ (the zero iso-contour), which is the constraint on the mass conservation.
In the 1D case illustrated in Fig. 3.6a, a trivial solution exists to reconstruct the signed distance without moving Γ and with $|\nabla\phi| = 1$ constraint. However, in a multidimensional case represented in Fig. 3.6b, imposing $|\nabla\phi| = 1$ on the closest nodes to the interface will lead to a shift of Γ. Then, a choice has to be made between these two constraints:



(a) *1D case* **(b)** *2D case*

Figure 3.6: *Illustration of interface displacement after redistanciation (straight line corresponds to ϕ before redistanciation and dashed line corresponds to ϕ after redistanciation)*

- Consider the closest nodes to the interface as exact values of the signed distance, which are not updated in the redistanciation process. This will lead to conservation of mass, but the loss of the $|\nabla\phi| = 1$ property on these nodes.
- Apply the redistanciation over all nodes, including the closest to the interface. This will lead to movements of the interface, but the $|\nabla\phi| = 1$ property will be true everywhere.

To reduce this discrepancy, some authors [185] tried to modify Eq. (3.35) by adding a source term counter-balancing the distortion from velocity. However, a redistanciation procedure is still necessary to enforce Eq. (3.36) after some iterations.

3.2.1.1 Original PDE redistanciation

The first approach is to consider Eq. (3.37) and the Hamiltonian is discretized using an upwind Godunov approach, such that the discrete Hamiltonian $\mathcal{H}_G$ in 2D (3D extension is straightforward) is defined as

$$\mathcal{H}_G = \begin{cases} \sqrt{\max\left(|\frac{\partial\phi}{\partial x}^-|_+^2, |\frac{\partial\phi}{\partial x}^+|_-^2\right) + \max\left(|\frac{\partial\phi}{\partial y}^-|_+^2, |\frac{\partial\phi}{\partial y}^+|_-^2\right)} - 1 & \text{if } \phi_0 \geq 0 \\ \sqrt{\max\left(|\frac{\partial\phi}{\partial x}^-|_-^2, |\frac{\partial\phi}{\partial x}^+|_+^2\right) + \max\left(|\frac{\partial\phi}{\partial y}^-|_-^2, |\frac{\partial\phi}{\partial y}^+|_+^2\right)} - 1 & \text{if } \phi_0 < 0 \end{cases}, \quad (3.38)$$

with $|a|_+ = \max(a, 0)$ the positive part of a, $|a|_- = \min(a, 0)$ the negative part of a, $\frac{\partial a}{\partial x}^+$ the biased derivative of a to the left and $\frac{\partial a}{\partial x}^-$ the biased derivative of a to the right. Note that other Hamiltonian discretizations are also possible, such as Lax Friedrichs or Roe [92]. Traditionally, the HJ-WENO5 approach of [92] is used for the gradient discretization with an RK2 or RK3 scheme for temporal integration. It has been shown later that this discretization leads to a second-order distance function [67] and the displacement of the zero iso-contour for ϕ. The reason of this low-order reinitialization compared to the high-order gradient computation is explained in [184]: high-order upwind schemes are not truly upwind as they imply values in the wrong side of the domain near the interface (values in Ω_g

are updated using ϕ in Ω_l and vice versa). Such an approach suffers from a limited order of accuracy and a loss of mass conservation while it is fairly easy to implement. Moreover, the value of the closest nodes to the interface can oscillate around $\phi = 0$ switching sign multiple times, preventing Eq. (3.37) from reaching a steady state. However, this behaviour can be highly diminished by introducing a smooth version of the sign distance $S(\phi_0)$, in [214] this is defined as

$$S(\phi_0) = \frac{\phi_0}{\sqrt{\phi_0^2 + \Delta x^2}} \quad . \tag{3.39}$$

3.2.1.2 Subcell fix approach

To solve the above problems, a first and second-order subcell fix was first introduced in [184], and a higher-order extension was proposed in [54]. The idea behind the subcell fix is to consider a variable stencil in the cells belonging to Ω_Γ such that they are not modified during the iteration process. This lead to a modification of Eq. (3.37)

$$\frac{\partial \phi}{\partial \tau} = \begin{cases} -\frac{1}{\Delta x}\left(\text{sign}(\phi_0)|\phi| - d\right) & \text{if } \mathcal{C} \in \Omega_\Gamma \\ -\text{sign}(\phi_0)\mathcal{H}_G(\phi) & \text{else} \end{cases} \quad , \tag{3.40}$$

with d the distance to the interface computed using the close nodes. Initially, this distance d is computed as a linear interpolation from neighbouring cells. This method allows Eq. (3.37) to reach a steady-state. Then, the accuracy of the method does not depend on the number of iterations anymore but on the initial discretization of the nodes close to the interface.

3.2.1.3 Interface preserving method

Sussman and Fatemi [211] proposed to add a constraint term in Eq. (3.37) to enforce mass conservation

$$\frac{\partial \phi}{\partial \tau} = \mathcal{H}\left(\phi, \nabla\phi\right) + \lambda g(\phi) \quad , \tag{3.41}$$

with

$$g(\phi) = H'(\phi)|\nabla\phi| \quad , \tag{3.42}$$

and

$$\lambda = -\frac{\int_\Omega H'(\phi)\mathcal{H}\left(\phi, \nabla\phi\right)}{\int_\Omega H'(\phi) g(\phi)} \quad . \tag{3.43}$$

This term was derived to ensure

$$\frac{\partial \int_\Omega H(\phi)}{\partial \tau} = 0 \quad , \tag{3.44}$$

which means that the volume included in the zero-isocontour does not vary in pseudo-time of the redistanciation equation. The constraint term discretization requires spatial numerical integration in computational cells and temporal derivatives. Hence the method is not straightforward to implement. Moreover, it does not lead to significant improvements in interface capturing accuracy even if it shows better conservation properties.

3.2.1.4 Constrained reinitialization

A more recent approach proposed by Hartmann et al. [78] also add a source term in the reinitialization step

$$\frac{\partial \phi}{\partial \tau} = \mathcal{H}\left(\phi, \nabla\phi\right) + \beta F_\phi \quad , \tag{3.45}$$

with $\beta = 0.5$ and $\mathcal{F}(\phi_0, \phi)$ a forcing term constructed such that the displacement of interface points is zero

$$\mathcal{F}_i(\phi_0, \phi) = \frac{1}{\Delta x}\left(-\phi_i + \frac{\phi_{i,0}}{\sum\limits_{j\in I_\Gamma}\phi_{0,j}} \sum_{j\in I_\Gamma} \phi_j\right) \quad , \tag{3.46}$$

with I_Γ the set of indices of neighbours cell belonging to the interface.
The constraint term allows to use the conventional HJ-WENO5 discretization for $\mathcal{H}\left(\phi, \nabla\phi\right)$ while fixing the points close to the interface in a high order manner.

3.2.1.5 Fast Marching Method

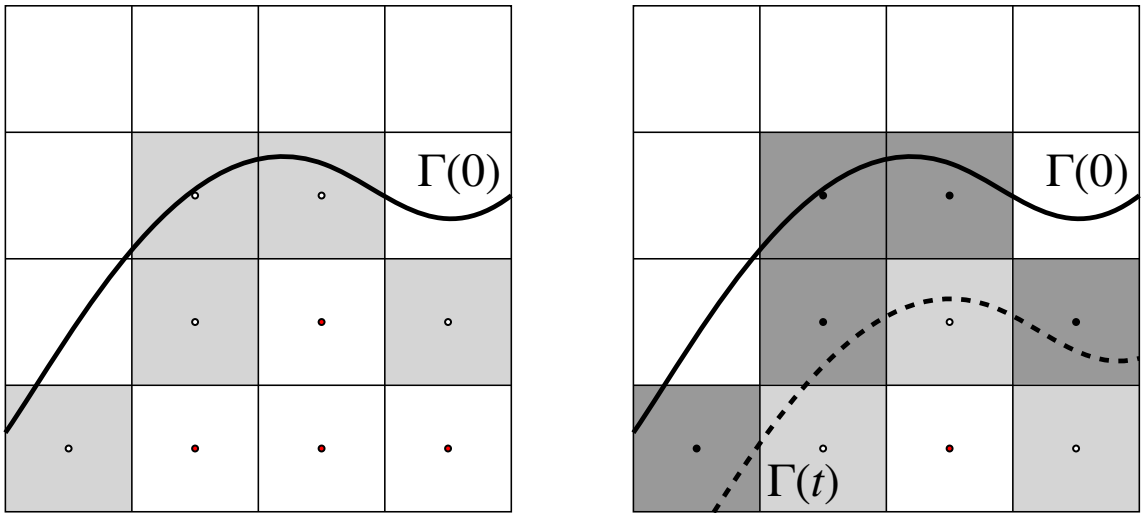
The other class of redistanciation methods considers the static version of the Hamilton-Jacobi equation. This leads to the stationary Eikonal equation with a unity normal speed and boundary condition on the zero iso-contour

$$\begin{cases} |\nabla\phi| = 1 & \mathbf{x} \in \Omega_l \cup \Omega_g \\ \phi = 0 & \mathbf{x} \in \Gamma \end{cases} \quad . \tag{3.47}$$

Eq. (3.47) can be solved efficiently using a Fast Marching Method (FMM) [197]. FMM relies on an upwind front evolving normal to the interface from Γ to Ω_l and Ω_g. Three list of nodes are defined: accepted (A), close (C) and far (F) nodes (in black, white and red respectively on Fig. 3.7). The accepted nodes are the one already behind the evolving front $\Gamma(t)$ while the close nodes are the direct neighbours of the accepted one and the far nodes are all the other nodes.
For a node to be computed at cell $\mathcal{C}$, one need to solve the following Godunov approximation of $\nabla\phi$ which leads to a quadratic equation in $\phi_\mathcal{C}$

$$\max\left(\frac{\partial\phi}{\partial x}^-, -\frac{\partial\phi}{\partial x}^+, 0\right) + \max\left(\frac{\partial\phi}{\partial y}^-, -\frac{\partial\phi}{\partial y}^+, 0\right) - 1 = 0 \quad , \tag{3.48}$$

with $\frac{\partial\phi}{\partial x}^-$ and $\frac{\partial\phi}{\partial x}^+$ the left and right biased first-order approximation of the ϕ derivative respectively. Note that higher-order upwind derivatives can also be used if accepted values are available on the stencil (see [180] for details). It is essential to use only nodes in the A list for the computation of the derivatives. To ensure it, nodes belonging to C and F lists are set to ∞ so they do not participate in the computation of Eq. (3.48).
There are two roots for the above equation, and a value of ϕ has to be chosen between them. In most of the case, one root will be smaller than the neighbour values and the other



(a) *Nodes at $t = 0$* **(b)** *Nodes at a time t*

Figure 3.7: *Illustration of the nodes definition in a FMM algorithm*

higher. In this case, causality implies picking the largest value.
In the updating of the accepted list, one has to be very careful not to violate causality. This is done by always taking the close node with the smallest value in the domain as the candidate to become accepted. The crucial point for FMM efficiency is quick access to the smallest value in the C list. This is done by the use of a heap sort approach.
The FMM algorithm can be sum up as:

1. Initialize the list of close and accepted node lists C and A
2. Perform a heap sort on C
3. Remove C(1) from the close list and add it in A
4. Add the adjacent nodes of C(1) belonging to F in C
5. Compute the new close values using Eq. (3.48)
6. Repeat step 2. to 5. until all nodes are accepted

As for the PDE-based approach, one can choose to conserve the iso-contour or to impose Eq. (3.36) even between values of different domains. This is done in the initialization of the close and accepted cells. If one want to conserve the zero iso-contour, then the closest nodes at $t = 0$ in Fig. 3.7a are considered already accepted and will not be modified in the FMM process. If one wants to recompute the nodes close to the interface in order to respect Eq. (3.36), then this can be done by solving Eq. (3.48) considering only interface positions as accepted points. These interface positions are computed from linear approximation.

3.2.1.6 Additional considerations on redistanciation

In order to choose a reinitialization method, some aspects have to be compared, such as accuracy, conservation and implementation efforts.

Accuracy One important aspect is the accuracy of the redistanciation as it can be limiting to achieve high order accuracy. A redistanciation that is not accurate jeopardizes the main advantage of SLS, which is to provide a high order representation of the interface for topology computations.
A complete comparison of redistanciation schemes based on the Hamilton-Jacobi equation is available in [203]. In fact, all the PDE methods but the subcell fix approach can lead to comparable accuracy with improvement for the Interface preserving method. Moreover, the use of a subcell fix is leading to higher parasitic currents, which could be due to the low-order reconstruction close to the interface.
High order FMM approaches are not straightforward and would require more implementation and computational efforts. As FMM is employed only for its efficiency, it is then not reasonable to consider higher-order implementations. Then, FMM can be considered as a second-order method for redistancing.

Conservation For conservation, the study of [203] showed that the Constrained reinitialization and the Interface preserving method were improving the mass conservation compared to the original method of [92]. However, the study focuses on evaluating sophisticated redistanciation methods and the original discretization is not considered in most of the test cases displayed. It is then not clear if the Constrained reinitialization and the Interface preserving methods show substantial improvement in conservation compared to the original discretization.

Implementation effort Finally, the implementation effort is noticeably unequal between the above-presented method. Indeed, the parallel implementation of an FMM algorithm is very tedious as it requires to respect causality. Several approaches has been proposed for this: Herrmann [80] and later McCaslin et al. [127] proposed to advance each front independently in each processor and use a rollback when a neighbour processor share a new accepted value. This ensures that causality is preserved and allows the front to propagate independently in each processor if they do not update shared values. In [233], the advancement is performed on a narrow band with a free width parameter. If the width is set to 0, the algorithm works as the serial one. When the width is set to ∞, all processors advance the front independently. A proper width allows to be efficient and respect causality.
For the PDE-based implementations, the original high order discretization is straightforward, whereas additional efforts are required for the Interface preserving method, and even more efforts are required for the Constrained reinitialization involving numerical integral evaluations.

From the above considerations, the original PDE redistanciation of [92] seems to provide a straightforward high order method in the vein of the SLS transport. Moreover, it is still used in modern solvers for evaporation applications [183] proving its relevance.

3.2.2 Flux computation

Classical advection schemes can be applied to ϕ as it is a fairly smooth function. However, discretization has a significant impact on mass conservation as numerical diffusion leads to artificial mass loss. This subject has been widely explored, and it has been shown multiple times that Level-Set performs well with high-order schemes. A complete comparison of ENO, WENO and HOUC schemes of different orders has been made in [149]. It is also possible to improve the method by using high-order Discontinuous Galerkin methods [49] or Gradient Augmented Level-Set approach [143]. Some semi-Lagrangian techniques have been proposed to remove the stability constraints [134; 226].
This is also a relevant choice as it is the most popular in the literature to solve such an equation. The most used scheme to compute the fluxes of ϕ is the WENO5 scheme of [93] as it provides a straightforward and accurate transport of ϕ

3.3 Conservative Level-Set method

In order to solve the mass conservation problems of the standard Level-Set method, a new formulation has been introduced by Olsson and Kreiss [150]. They started from the Heaviside definition of Sussman et al. [214]

$$H_\Gamma^\epsilon = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ 1 & \text{if } \phi > \epsilon \\ 1 + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & \text{otherwise} \end{cases} , \tag{3.49}$$

with ϵ half the thickness of the interface.
By noticing that

$$V(H_\Gamma^\epsilon = 0.5) \approx \int_\Omega H_\Gamma^\epsilon \quad , \tag{3.50}$$

with $V(H_\Gamma^\epsilon = 0.5)$ the mass contained in the isocontour $H_\Gamma^\epsilon = 0.5$. In fact, in the limit $\epsilon \to 0$, $V(H_\Gamma^\epsilon = 0.5) = \int_\Omega H_\Gamma^\epsilon$.
Olsson and Kreiss argued that transporting a function $\psi = H_\epsilon$ instead of the standard signed distance ϕ would highly increase the mass conservation properties. In the lastest versions of the method, the definition of the function ψ is defined in a continuous manner as

$$\psi = \frac{1}{2}\left(\tanh\left(\frac{\phi}{2\epsilon}\right)\right) \quad . \tag{3.51}$$

The transport equation for this new function is then

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma \psi) = 0 \quad . \tag{3.52}$$

3.3.1 Reinitialization

For the same reasons that ϕ cannot maintain the property $|\nabla\phi| = 1$ during transport, there are no guarantee that the hyperbolic tangent profile ψ will remain unchanged. This takes the form of local modifications of the interface thickness ϵ which would lead to a bad representation of the interface and topology computation from it.
An additional conservation equation has to be solved in pseudo time to overcome this problem

$$\frac{\partial\psi}{\partial\tau} + \nabla\cdot\mathbf{F}(\psi) = 0 \quad . \tag{3.53}$$

Note that it is imperative to write the reinitialization in a conservative way in order to keep the exact conservation of ψ. The "conservative" aspect of the method is only true for the transported variable ψ and not the mass itself. This is because the thickness ϵ cannot be too small in order to be resolvable on the computational mesh. Usually, $\epsilon = 0.5\Delta x$.
This flux $\mathbf{F}$ is composed of a compression flux and a diffusive flux which will tend to retrieve a hyperbolic tangent with an interface thickness ϵ.
Therefore, Eq. (3.53) is purely artificial and only corresponds to a numerical need to maintain the interface thickness uniform and constant in time.

3.3.1.1 Original reinitialization

In the original paper, the compression flux is defined as

$$\mathbf{F}_{comp} = \psi(1-\psi)\mathbf{n} \quad , \tag{3.54}$$

with $\mathbf{n}$ computed from ψ as

$$\mathbf{n} = \frac{\nabla\psi}{|\nabla\psi|} \quad . \tag{3.55}$$

This flux acts as a resharpening in the normal direction of the interface in the regions where $0 < \psi < 1$.
In order to keep the interface thickness into a resolvable thickness ϵ on the mesh, a small diffusive flux is added

$$\mathbf{F}_{diff} = \epsilon\nabla\psi \quad , \tag{3.56}$$

then Eq. (3.53) can be written as

$$\frac{\partial\psi}{\partial\tau} + \nabla\cdot(\psi(1-\psi)\mathbf{n}) = \nabla\cdot(\epsilon\nabla\psi) \quad . \tag{3.57}$$

With these definitions, the compression flux acts in the normal direction while the diffusion flux acts in both normal and tangential directions. The normal diffusion is expected to be counter-balanced by the compression. However, the tangential diffusion does not have a compression counterpart, which can lead to interface displacements.

3.3.1.2 Diffusion in the reinitialization

This is why in the second part of their paper [151], Olsson et al. modified the diffusive flux given in Eq. (3.56)

$$\mathbf{F}_{diff} = \epsilon \left(\nabla\psi \cdot \mathbf{n}\right) \mathbf{n} \quad , \tag{3.58}$$

leading to the reinitialization equation

$$\frac{\partial\psi}{\partial\tau} + \nabla \cdot (\psi(1-\psi)\mathbf{n}) = \nabla \cdot \left(\epsilon \left(\nabla\psi \cdot \mathbf{n}\right) \mathbf{n}\right) \quad . \tag{3.59}$$

It can be shown that Eq. (3.51) satisfies

$$\frac{\partial\psi}{\partial\phi} = \frac{\psi\left(1-\psi\right)}{\epsilon} \quad , \tag{3.60}$$

which is compatible with the steady state of Eq. (3.59)

3.3.1.3 Normal computation in the reinitialization

As pointed out by Desjardins et al. in [48], even if the advection and reinitialization are accurate and TVD, the ψ function can deviate locally from a hyperbolic tangent, which can introduce spurious normal computation if Eq. (3.55) is used.
They proposed an Accurate Conservative Level Set (ACLS) method in which they recompute a distance function ϕ_{FMM} from the isocontour $\psi = 0.5$ and compute the normal with

$$\mathbf{n} = \frac{\nabla\phi_{FMM}}{|\nabla\phi_{FMM}|} \quad . \tag{3.61}$$

This new approach alleviates the TVD constraint on the advection scheme to solve Eq. (3.52). This is why HOUC5 schemes can be employed in the ACLS method.

3.3.1.4 Interface displacement in the reinitialization

Even if the ACLS approach solves some issues of SLS, the reinitialization still suffers from numerical errors tending to deform the interface. These errors do not show up when the interface is in a highly dynamic flow. However, it can be predominant in regions of the domain where the interface is still.

Localized reinitialization This problem has been widely investigated, and one solution would be to apply reinitialization only in regions where it is required. This has been done in [189] by rewriting Eq. (3.59) as

$$\frac{\partial\psi}{\partial\tau} + \alpha\nabla \cdot (\psi(1-\psi)\mathbf{n}) = \alpha\nabla \cdot \left(\epsilon \left(\nabla\psi \cdot \mathbf{n}\right) \mathbf{n}\right) \quad . \tag{3.62}$$

Hence, if $\alpha = 0$, no reinitialization is performed, while $\alpha = 1$ corresponds to the original method. This parameter is local and is defined based on numerical experiments.
McCaslin et al. [128] proposed another modification of the reinitialization equation

$$\frac{\partial \psi}{\partial \tau} + \nabla \cdot (\alpha \psi (1-\psi)\mathbf{n}) = \nabla \cdot (\alpha \epsilon \left(\nabla \psi \cdot \mathbf{n}\right)\mathbf{n}) \quad . \tag{3.63}$$

This time, the original reinitialization is retrieved if $\nabla \alpha \cdot \mathbf{n}$. The parameter α is based on flow kinematics and the numerical diffusion of ψ when solving Eq. (3.52).
Other attempts to solve this problem has been proposed in [225].

A reinitialization reformulation In [38], Chiodi et al. proposed a new reformulation of the reinitialization leading to a huge reduction of interface deformation due to reinitialization.
The idea is to rewrite fluxes by using trigonometric identities (all the details can be found in Sahut PhD thesis [186]) and a mapping of ϕ

$$\mathbf{F}_{comp} = \psi\left(1-\psi\right)\mathbf{n} = \frac{1}{4\cosh^2\left(\frac{\phi_{map}}{2\epsilon}\right)}\mathbf{n} \quad , \tag{3.64}$$

$$\mathbf{F}_{diff} = \epsilon\left(\nabla\psi \cdot \mathbf{n}\right)\mathbf{n} = \left(\frac{1}{4\cosh^2\left(\frac{\phi_{map}}{2\epsilon}\right)}|\nabla\phi \cdot \mathbf{n}|\right)\mathbf{n} \quad , \tag{3.65}$$

with the ϕ_{map} the ϕ mapping defined as

$$\phi_{map} = \epsilon \log\left(\frac{\psi}{1-\psi}\right) \quad . \tag{3.66}$$

This new definition of the compression flux leads to the following reinitialization equation

$$\frac{\partial \psi}{\partial \tau} = \nabla \cdot \left(\frac{1}{4\cosh^2\left(\frac{\phi_{map}}{2\epsilon}\right)}\left(|\nabla\phi_{map} \cdot \mathbf{n}| - 1\right)\mathbf{n}\right) \quad . \tag{3.67}$$

The mapping of ϕ_{map} does not remove potential oscillatory behaviour in the normal components, this is why the normal are still computed from Eq. (3.61).
Note that is in order to compute ϕ_{map} in Eq. (3.66), the value of ψ needs to be bounded between $]\eta_c; 1-\eta_c[$ with η_c an arbitrary small number. To avoid any important clipping of ψ, the HOUC5 scheme and any other oscillating schemes cannot be used anymore, and bounded schemes such as BQUICK or BHOUC is preferred [82].
This method has shown to be as accurate as the original ACLS method but with the advantage of not moving the interface in still regions.

The locking fix Using the formulation of Eq. (3.67), some grid locking can occur when two interface fronts are close to each other. This is explained looking at the term $(|\nabla\phi_{map} \cdot \mathbf{n}| - 1)$. When two fronts are close to each other, ϕ_{map} is close to zero, and the flux is dominated by

the compressive term (the 1 in the appearing in the flux). This flux is then tremendous and not balanced anymore by the diffusion, which creates this locking phenomenon. In [89], a new formulation is proposed, written as $(\nabla\phi_{map} \cdot \mathbf{n} - \mathbf{n} \cdot \mathbf{n})$ with the normal computed from $\mathbf{n} = \nabla\phi_{FMM}$ without normalization.

3.3.1.5 Comparison of the reinitialization methods

For the following test case, reinitialization is performed on a sphere with a resolution $D/\Delta x = 20$ as in [38]. The goal is to compare the original reinitialization Eq. (3.57) with the ACLS reinitialization Eq. (3.59) and the Chiodi reinitialization Eq. (3.67). Note that the fix proposed in [89] has no impact on this test case as the terms are different only in front merging configurations.
In Figs. 3.8 to 3.10 are compared three different states of the sphere: the initial shape, the shape after 100 iterations of reinitialization steps and the shape after 1000 reinitialization steps. It is clear that modifying the direction of the diffusion by introducing only diffusion in the normal direction to the interface reduces the distortion of the droplet by comparing Figs. 3.8 and 3.9. However, the reinitialization Eq. (3.59) still deforms the sphere by flattening the edges.

(a) *Initial shape* **(b)** *100 iterations* **(c)** *1000 iterations*

Figure 3.8: *Effect of the reinitialization from Eq. (3.57) on a static droplet*

(a) *Initial shape* **(b)** *100 iterations* **(c)** *1000 iterations*

Figure 3.9: *Effect of the reinitialization from Eq. (3.59) on a static droplet*

Finally, the modification of Chiodi allows to keep a perfect sphere even after 1000 iterations as seen in Fig. 3.10. This last result is very convincing for applications such as isolated droplets evaporating.

(a) *Initial shape* **(b)** *100 iterations* **(c)** *1000 iterations*

Figure 3.10: *Effect of the reinitialization from Eq. (3.67) on a static droplet*

This simple test case shows that the reinitialization taking into account both the Chiodi modification and the locking fix of [89] is the most-suited method for accurate and robust reinitialization of ψ. This takes the following form

$$\frac{\partial \psi}{\partial \tau} = \nabla \cdot \left(\frac{1}{4\cosh^2\left(\frac{\phi_{map}}{2\epsilon}\right)} \left(\nabla \phi_{map} \cdot \mathbf{n} - \mathbf{n} \cdot \mathbf{n}\right) \mathbf{n} \right) \quad , \tag{3.68}$$

with the normal obtained with $\mathbf{n} = \nabla \phi_{FMM}$.

3.3.2 Flux computation

As for the SLS, classical advection schemes can be applied to ψ.
In the original paper [150], Olsson and Kreiss investigated a large range of second-order TVD methods based on upwind schemes with piecewise linear reconstruction and retained a MUSCL scheme with superbee limiter (see [112] for more details).
In the second part of the article [151] they employed a finite element discretization to improve the accuracy of the advection step.
Later, Desjardins et al. [48] proposed to use higher-order finite volume schemes and compared HOUC5 [46] with WENO5. They concluded that using HOUC5 were giving more accurate results.
A Discontinous Galerkin method has also been presented in [155] which improves accuracy for the transport test cases shown in their work.
The flux reconstruction can be chosen using the following considerations:

- The transport and conservation properties are improved by high accuracy.
- The ψ field is allowed to be spurious as topological properties are computed from ϕ_{FMM}.
- The ψ function needs to be bounded as explained in Section 3.3.1.4.

A BHOUC5 reconstruction [82] seems to be the most-suited flux reconstruction to reach all these requirements in a finite-volume framework. It is more dispersive compared to a WENO5 scheme, however, it is less diffusive, which is expected to improve the transport accuracy. Moreover, the scheme remains bounded as upwind fluxes are used when ψ experiences undershoot or overshoot.

3.4 Coupled Level-Set Volume-of-Fluid method

The coupling between LS and VOF was first introduced by Sussman and Puckett [213] to simulate microscale jetting devices. This application needs an accurate computation of the curvature for surface tension contribution and enough mass conservation to capture jet's tail before break up. Based on SLS algorithm, they observed that the tail disappears because of mass loss while VOF algorithm was introducing oscillatory behaviours because of the inaccuracies in curvature computation from f. The idea is then is to transport both f and ϕ and couple them through the PLIC reconstruction of the interface and the reinitialization of ϕ. This results in solving Eq. (3.2) twice with $c = f, \phi$

$$\frac{\partial \phi}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma \phi) = 0 \quad , \tag{3.69}$$

$$\frac{\partial f}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma f) = 0 \quad . \tag{3.70}$$

Note that another class of coupled LS-VOF method was also introduced with both algebraic [3] or geometric [208] VOF where only f is transported while ϕ is deduced from the $f^{-1}(0.5)$ iso-contour. However, this approach was mainly used to retrieve more accurate curvature and are not considered here as there are not completely coupled methods, as ϕ does not have an evolution equation and is not used in the PLIC reconstruction.

3.4.1 Coupling between Level-Set and Volume-Of-Fluid

3.4.1.1 Normal evaluation

In CLSVOF, the interface normal need to be retrieved as for a classical geometric VOF algorithm. The idea is to use ϕ instead of f to compute this quantity in order to circumvent the issues described in Section 3.1.1.1. Hereafter, three strategies are presented.

Plane fit The original approach of Sussman and Puckett [213] used a plane fit approach written in the Least-Square sense. The plane is reconstructed such that the error between the distance to the interface ϕ and the distance to the plane is minimized in a 3-stencil around the considered cell. For a 2D mesh cell $\mathcal{C}_{i,j}$, this gives the following expression for the error $E_{i,j}$ to minimize

$$E_{i,j} = \sum_{k=i-1}^{i+1} \sum_{l=j-1}^{j+1} w_{k,l} \left(\phi_{k,l} - \mathbf{n}_{i,j} \cdot \mathbf{\Delta x}_{k,l} + d_{i,j} \right) \quad , \tag{3.71}$$

with $w_{k,l}$ a weight designed to localize the system of equation around $\mathbf{x}_{i,j}$ and $\mathbf{\Delta x}_{k,l}$ the displacement from $\mathbf{x}_{k,l}$ to $\mathbf{x}_{i,j}$.
In [91] the weights are computed using

$$w_{k,l} = \begin{cases} 10^{-3}\left(\frac{1}{22}\right)^{|k|+|l|} & \text{if } |\phi_{k,l}| > \epsilon \\ \left(1+\cos\left(\frac{\pi\phi_{k,l}}{\epsilon}\right)+10^{-3}\right)\left(\frac{1}{22}\right)^{|k|+|l|} & \text{if } |\phi_{k,l}| \leq \epsilon \end{cases} \quad . \tag{3.72}$$

The minimization requires to invert a 3×3 (4×4 in 3D) matrix and leads to a second-order accurate computation of the normal. Note that the value of d computed from this linear system does not ensure that the PLIC reconstruction respects the volume of the cell. It is then recomputed based on $\mathbf{n}_{i,j}$ and $f_{i,j}$ as described in Section 3.1.1.2.
In [218], authors noticed that using the full 3×3 block of ϕ values could lead to numerical errors when two interface fronts were crossing the stencil. In this case, they proposed to use a variable stencil to consider only points with a ϕ value associated with the correct interface piece.

Standard derivation The plane fit method gives an accurate method for normal computation. However it requires to solve a linear system and dynamic stencil if the correction of [218] is included. The interest of such a method is then arguable compared to algorithms based on f presented in Section 3.1.1.1. This is why, several works [205; 238] use centered finite differences to retrieve the normal from

$$\mathbf{n} = -\frac{\nabla\phi}{|\nabla\phi|} \quad . \tag{3.73}$$

This expression leads to an efficient second-order accurate approximation of the interface.

CLSMOF As for the MOF algorithm, some CLSVOF algorithms [91; 139] also transport the center of mass of the fluid and switch between a normal computation based on ϕ or on MOF depending on an Interface Resolution Quality IRQ defined as

$$IRQ = \frac{1}{\Delta x|\kappa|} \quad , \tag{3.74}$$

with κ the local curvature. This methodology has been compared to standard CLSVOF on airblast atomization [141] and a huge impact on the atomization process was observed. While the CLSVOF method led to predominant sheet breakup, the CLSMOF was giving predominant ligament breakup instead.

The main asset of using a CLSVOF approach instead of VOF is a simple and accurate method to compute the normal. Then, the standard derivation of Eq. (3.73) constitutes the most relevant choice here as it does not require solving a linear system or transport additional information without a significant loss of accuracy in the target applications.

3.4.1.2 Redistanciation

Another coupling is done through the redistanciation of ϕ after transport. Instead of solving Eq. (3.37) or Eq. (3.47), the CLSVOF redistanciation step is based on f to enforce mass conservation. The idea is to give a new zero iso-contour for the ϕ field corresponding to the exact distance to the PLIC reconstruction. In the manuscript, the signed distance ϕ obtained from the CLSVOF method will be renamed as ϕ_{PLIC}.

Reset approach In the original CLSVOF method [213], the LS was completely reset in a narrow band of 5 cells around the interface to match exactly the distance to the PLIC reconstruction. This was done by using the PLIC reconstruction of all neighbouring cells and chose the shortest distance to one of the planes. One drawback of this approach is that the high accuracy of ϕ is lost through the reconstruction process, ϕ_{PLIC} contains errors from the linear reconstructions of the interface.

Mixed approach In [218], the reset is only applied in the mixed cells (where $0 < f < 1$) using the fact that the distance to the plane is simply $\phi_{PLIC} = d$. However, in all other cells, Eq. (3.37) was solved with the HJ-WENO5 scheme. This algorithm allows higher accuracy while keeping the anchor of the PLIC reconstruction distance at interface cells.

Relaxation Finally, Le Chenadec and Pitsch [36] noticed that using the reset approach at each timestep led to spurious behaviour of the interface evolution. They proposed to introduce a relaxation to weaken the coupling between f and ϕ

$$\phi_{PLIC} = \omega\phi + (1-\omega)d \quad , \tag{3.75}$$

with ω the relaxation weight. Their study led to the following definition of ω as a function of the discrepancy between ϕ and d

$$\omega = \exp\left(-\alpha \max\left(\frac{|\phi - d|}{\eta} - 1, 0\right)^2\right) \quad , \tag{3.76}$$

with α a relaxation speed set to 10, and η a cutoff scale below which the discrepancy is considered negligible (taken as $0.01\Delta x$).

A combination: HJ-relax To keep the simplicity of the redistanciation and mitigate the errors induced in the ϕ function close to the interface, a combination of the mixed approach [218] and the relaxation of [36] is proposed, referred as HJ-relax. Then, ϕ reset is only performed in mixed cells, including the relaxation, while the HJ-WENO5 scheme is used otherwise.

3.4.1.3 VOF truncation

Most of the CLSVOF approaches also alleviate the mass conservation constraint by cleaning the field f using ϕ to avoid any jetsam or flotsam resulting from the transport step. In

atomization processes, where a lot of under-resolved structures appear, the presence of jetsam or flotsam can lead to robustness issues.
The VOF truncation is done either considering that if $|\phi| > \Delta x$ then the cell cannot contain an interface and f is set to 0 or 1 depending on the sign of ϕ [213; 205; 218]. Another way to restrain the VOF proposed in [139], is to check if ϕ experiences any sign-change with its neighbour. f is set to 0 or 1 if no sign-change is experienced in ϕ.

3.4.2 Flux computation

It is essential to use a geometric flux reconstruction for VOF to retrieve the desired properties discussed in Section 3.1.2. As ϕ is required for PLIC reconstruction, it is also important to have consistency between the time integration of f and ϕ. A discrepancy between the two quantities could lead to an inaccurate normal approximation which totally jeopardizes one of the main advantages of coupling LS and VOF. Then, if a dimensional-splitting algorithm is used for VOF, LS need also to be treated in the same way. As discussed previously, the 1D equation to solve for ϕ should be of the form Eq. (3.15)

$$\frac{\partial f}{\partial t} + \frac{\partial \left(u_s f\right)}{\partial x_s} = f\frac{\partial u_s}{\partial x_s} \quad , \tag{3.77}$$

$$\frac{\partial \phi}{\partial t} + \frac{\partial \left(u_s \phi\right)}{\partial x_s} = \phi\frac{\partial u_s}{\partial x_s} \quad , \tag{3.78}$$

with Eq. (3.78) solved using classical finite volume schemes.
In [36], an unsplit VOF scheme is used and the LS can be advanced as described in Section 3.2.

3.5 Conclusion

In this chapter, a review of interface capturing methods was provided with a focus on the four most popular methods of the literature. They all present strengths and weaknesses for the study of two-phase flows. While the VOF method is mass conservative, it suffers from limitations in accuracy because of the geometric nature of the flux. Another class of methods based on an implicit representation of the interface led to the SLS method, which provides an arbitrary accurate interface representation while it suffers from mass conservation. The CLS method was introduced to improve the mass conservation properties while keeping an arbitrary accurate interface representation. Finally, the CLSVOF method uses both the mass conservation property of VOF and the accuracy of the LS representation.

The next chapter aims to provide a fair comparison between an up-to-date version of each method with the choices discussed in the different section and summarized in Table 3.2 to expose their numerical properties on canonical test cases.

method	VOF	SLS	ACLS	CLSVOF
$F^{(c)}$	WY	WENO5	BHOUC5	WY-WENO5
Reconstruction	PLIC	–	–	PLIC
Reinitialization	–	HJ-WENO5	Eq. (3.68)	HJ-relax
$\mathbf{n}$	ELVIRA	Eq. (3.73)	Eq. (3.61)	Eq. (3.73)

Table 3.2: *Summary of computation choices for interface capturing methods*

Chapter 4

Evaluation of interface capturing methods

Contents

As the research in interface capturing methods is highly active, updating comparison studies is required to take advantage of the recent advances. The previous chapter provided an overview of available methods and their associated algorithms. From the state-of-the-art, four interface capturing methods were selected for their proven capabilities to transport an interface in a sharp manner.

In this chapter, the implementation of these methods is given in Section 4.1 with details on the temporal and spatial discretizations. Then, a complete study is pursued in Section 4.3 to compare the interface capturing methods in both 2D and 3D configurations with imposed velocity fields. The comparison is based on shared metrics introduced in Section 4.2.

4.1 Interface capturing methods choice

Based on the choices of Section 3.5, the numerical discretization is detailed for each interface capturing method.

4.1.1 VOF

A directional step of the Weymouth and Yue scheme can be written as

$$f^*_{i,j,k} = f^n_{i,j,k} - \frac{\Delta t}{\Delta x}\left(F^{(f)}_{i+\frac{1}{2},j,k} - F^{(f)}_{i-\frac{1}{2},j,k}\right) + f_{c,i,j,k}\frac{\Delta t}{\Delta x}\left(u^n_{i+\frac{1}{2},j,k} - u^n_{i-\frac{1}{2},j,k}\right) \quad , \tag{4.1}$$

with $F^{(f)}_{i+\frac{1}{2},j,k}$ a geometric flux computed using the Eulerian Implicit approach.
The normal computation of the PLIC reconstruction is performed using the ELVIRA method keeping a compact $3\times3\times3$ stencil in 3D. Thus, the normal $\mathbf{n}$ does not achieve second order convergence for planes. However, it provides a good trade-off between accuracy, compactness and efficiency. The plane parameter d is retrieved using analytical expressions [193].
Finally, the f field is clipped after each directional advection. This step is required to avoid the effects of floating-point round-off errors from the geometric flux computation. f is set to 0 if $f < \eta_c$ and f is set to 1 if $f > 1-\eta_c$ where η_c is a cut-off threshold. In our simulations, it has been found that round of errors were superior in 3D due to the additional arithmetic operations required to compute the fluxes. η_c is then set to 10^{-12} by default and increased to 10^{-8} for 3D simulations.
The update of f is summarized in the following algorithm:

1. Set $f^s = f^n$
2. Compute the compression/dilatation factor f_c using Eq. (3.33) from f^n
3. Set $s = 1$ and repeat the following steps until $s = N_{dim}$
   (i) Perform a PLIC reconstruction by computing $\mathbf{n}$ and d in all cells were $0 < f^s < 1$
   (ii) Solve Eq. (4.1) in a given direction to find f^{s+1}
   (iii) Clip f^{s+1} using η_c

The direction order is alternately swapped at each timestep to mitigate the splitting error and retrieve a second order time integration (alternate Lie splitting). The swap is done using cyclic permutation.

4.1.2 SLS

The SLS algorithm retained in this manuscript has been chosen for its simplicity and efficiency. The transport equation is discretized as follows

$$\begin{aligned}\phi_{i,j,k}^{n+1} = \phi_{i,j,k}^{n} &- \frac{\Delta t}{\Delta x}\left(F^{(\phi)}_{i+\frac{1}{2},j,k} - F^{(\phi)}_{i-\frac{1}{2},j,k}\right) \\ &- \frac{\Delta t}{\Delta y}\left(F^{(\phi)}_{i,j+\frac{1}{2},k} - F^{(\phi)}_{i,j-\frac{1}{2},k}\right) \\ &- \frac{\Delta t}{\Delta z}\left(F^{(\phi)}_{i,j,k+\frac{1}{2}} - F^{(\phi)}_{i,j,k-\frac{1}{2}}\right) \quad ,\end{aligned} \tag{4.2}$$

with $F^{(\phi)}_{i+\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k}\bar{\phi}_{i+\frac{1}{2},j,k}$ the numerical flux.
$\bar{\phi}_{i+\frac{1}{2},j,k}$ is a reconstruction of ϕ at the cell face $\partial\mathcal{C}_{i+\frac{1}{2},j,k}$. A high-order WENO5 reconstruction [93] is performed. The time integration of Eq. (4.2) is performed using a SSP-RK2 scheme [199]

$$\begin{aligned}\phi^{(1)} &= \phi^{n} + \Delta t\mathcal{L}(\phi^{n}) \\ \phi^{n+1} &= \frac{1}{2}\phi^{n} + \frac{1}{2}\phi^{(1)} + \frac{1}{2}\Delta t\mathcal{L}(\phi^{(1)}) \quad ,\end{aligned} \tag{4.3}$$

with $\mathcal{L}(\cdot)$ the spatial operator.
Higher-order temporal discretization could also be used. However, only few improvements have been observed [216] compared to the corresponding computational time increase. More generally, Osher and Fedwick argued [152] that the accuracy of the transport equation was driven by the spatial discretization more than the time integration. The redistancing needs to solve the following equation in pseudo-time

$$\phi_{i,j,k}^{n+1} = \phi_{i,j,k}^{n} - \Delta\tau S_{i,j,k}\mathcal{H}_G(\phi^{n})_{i,j,k} = 0 \quad , \tag{4.4}$$

with $\mathcal{H}_G(\phi^{n})_{i,j,k}$ the Hamilton-Jacobi operator discretized using Eq. (3.38). The sign function $S_{i,j,k}$ is computed using Eq. (3.39)

$$S_{i,j,k} = \frac{\phi_{i,j,k}}{\sqrt{\phi_{i,j,k}^2 + \Delta x^2}} \quad . \tag{4.5}$$

In this operator, the first derivatives of ϕ are obtained with a WENO5 discretization proposed in [92]. The integration in time is performed using the same SSP-RK2 scheme. The pseudo-timestep is chosen to be $\Delta\tau = 0.5\Delta x$. Noticing that the interface displacement is bounded by the cell size, only two timesteps of the redistancing step are required in order to fix the signed distance [183] close to the interface.
The update of ϕ is summarized in the following algorithm:

1. Solve Eq. (4.2) to retrieve ϕ^* the signed distance before redistancing.
2. Compute the smooth signed distance S using Eq. (4.5) from ϕ^*
3. Perform 2 iterations of Eq. (4.4) to find ϕ^{n+1}

4.1.3 ACLS

The ACLS algorithm used in this work is based on the latest works of Janodet et al. [89], as it is the most advanced and documented method available in the literature. The interface width, ϵ is taken as $0.5\Delta x$. The transport equation is discretized as follows

$$\begin{aligned}\psi_{i,j,k}^{n+1} = \psi_{i,j,k}^{n} &- \frac{\Delta t}{\Delta x}\left(F^{(\psi)}_{i+\frac{1}{2},j,k} - F^{(\psi)}_{i-\frac{1}{2},j,k}\right) \\ &- \frac{\Delta t}{\Delta y}\left(F^{(\psi)}_{i,j+\frac{1}{2},k} - F^{(\psi)}_{i,j-\frac{1}{2},k}\right) \\ &- \frac{\Delta t}{\Delta z}\left(F^{(\psi)}_{i,j,k+\frac{1}{2}} - F^{(\psi)}_{i,j,k-\frac{1}{2}}\right) \quad , \end{aligned} \tag{4.6}$$

with $F^{(\psi)}_{i+\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k}\overline{\psi}_{i+\frac{1}{2},j,k}$ the numerical flux.
$\bar{\psi}_{i+\frac{1}{2},j,k}$ is a reconstruction of ψ at the cell face $\partial\mathcal{C}_{i+\frac{1}{2},j,k}$. A bounded high-order BHOU5 reconstruction [82] is performed here. The procedure can be explained trough the following algorithm:

1. Equation (4.6) is solved using $\overline{\psi}^{HOUC5}$ reconstructions.
2. If $\psi^{n+1}_{i,j,k} < 0$ or $\psi^{n+1}_{i,j,k} > 1$ then
   - All fluxes of cell $\mathcal{C}_{i,j,k}$ are updated using $\overline{\psi}^{UW}$ reconstructions
   - If a neighbouring cell shares a face with $\mathcal{C}_{i,j,k}$, then the flux is also updated with $\overline{\psi}^{UW}$ reconstructions
3. Equation (4.6) is solved again using the updated fluxes.

It is crucial to update fluxes of the neighbouring cells to keep the strict conservation properties of the finite volume framework. The integration is performed using a SSP-RK3 scheme [199] more suited for highly dispersive schemes such as HOUC5

$$\begin{aligned}\phi^{(1)} &= \phi^{n} + \Delta t\mathcal{L}(\phi^{n}) \\ \phi^{(2)} &= \frac{3}{4}\phi^{n} + \frac{1}{4}\phi^{(1)} + \frac{1}{4}\Delta t\mathcal{L}(\phi^{(1)}) \\ \phi^{n+1} &= \frac{1}{3}\phi^{n} + \frac{2}{3}\phi^{(2)} + \frac{2}{3}\Delta t\mathcal{L}(\phi^{(2)}) \quad . \end{aligned} \tag{4.7}$$

Then the reinitialization is written as

$$\begin{aligned}\psi_{i,j,k}^{n+1} = \psi_{i,j,k}^{n} &+ \frac{\Delta \tau}{\Delta x}\left(F^{(\psi)}_{i+\frac{1}{2},j,k} - F^{(\psi)}_{i-\frac{1}{2},j,k}\right) \\ &+ \frac{\Delta \tau}{\Delta y}\left(F^{(\psi)}_{i,j+\frac{1}{2},k} - F^{(\psi)}_{i,j-\frac{1}{2},k}\right) \\ &+ \frac{\Delta \tau}{\Delta z}\left(F^{(\psi)}_{i,j,k+\frac{1}{2}} - F^{(\psi)}_{i,j,k-\frac{1}{2}}\right) \quad , \end{aligned} \tag{4.8}$$

with $F^{(\psi)}_{i+\frac{1}{2},j,k}$ the reinitialization flux from [90] expressed as

$$F^{(\psi)}_{i+\frac{1}{2},j,k} = \frac{\left(\nabla\phi_{map}|_{i+\frac{1}{2},j,k} - \mathbf{n}|_{i+\frac{1}{2},j,k}\right)\cdot\mathbf{n}|_{i+\frac{1}{2},j,k}}{4\cosh^2\left(\frac{\phi_{map,i+\frac{1}{2},j,k}}{2\epsilon}\right)} n_{x,i+\frac{1}{2},j,k} \quad , \tag{4.9}$$

where $\phi_{map,i+\frac{1}{2},j,k}$ is simply defined as $\frac{1}{2}\left(\phi_{map,i,j,k} + \phi_{map,i,j,k}\right)$ and $\mathbf{n}|_{i+\frac{1}{2},j,k} = \nabla\phi_{FMM}|_{i+\frac{1}{2},j,k}$. Gradients $\nabla\phi_{map}|_{i+\frac{1}{2},j,k}$ and $\nabla\phi_{FMM}|_{i+\frac{1}{2},j,k}$ on the faces are computed using central differences.

The mapping ϕ_{map} is reconstructed following Eq. (3.66) with attention drawn to boundedness of ψ. The expression is used with a modified $\widetilde{\psi}$ such that it is defined in $]\eta_c; 1-\eta_c[$ with $\eta_c = 10^{-12}$. Note that we do not directly clip ψ but just use the clipped version $\widetilde{\psi}$ in Eq. (3.66).

The reconstruction of a signed distance ϕ_{FMM} from the isocontour $\psi^{-1}(0.5)$ is performed using the FMM described in Section 3.2.1.5. The pseudo-timestep $\Delta\tau$ is taken here as 0.5ϵ.

The update of ψ is summarized in the following algorithm:

1. Advance the interface by solving equation (3.1) to obtain ψ^* before reinitialization
2. Compute the signed distance ϕ_{FMM} from the isocontour $\psi^* = 0.5$ and ϕ_{map} from $\widetilde{\psi}^*$
3. Perform one iteration of Eq. (3.59) to obtain ψ^{n+1}

4.1.4 CLSVOF

Considering a cartesian-grid framework, the selected CLSVOF method is also based on dimensional-splitting. f is advanced using the same procedure described in Section 4.1.1. For the signed distance ϕ, a standard approach such as the one presented in Section 4.1.2 cannot be used as the time integration has to be consistent with the one used for f transport. Instead, the following equation is solved

$$\phi^*_{i,j,k} = \phi^n_{i,j,k} - \frac{\Delta t}{\Delta x}\left(F^{(\phi)}_{i+\frac{1}{2},j,k} - F^{(\phi)}_{i-\frac{1}{2},j,k}\right) + \phi^n_{i,j,k}\frac{\Delta t}{\Delta x}\left(u^n_{i+\frac{1}{2},j,k} - u^n_{i-\frac{1}{2},j,k}\right) \quad , \tag{4.10}$$

with $F^{(\phi)}_{i+\frac{1}{2},j,k} = u_{i+\frac{1}{2},j,k}\overline{\phi}_{i+\frac{1}{2},j,k}$ the numerical flux.

$\overline{\phi}_{i+\frac{1}{2},j,k}$ is a reconstruction of ϕ at the cell face $\partial\mathcal{C}_{i+\frac{1}{2},j,k}$. The same high-order WENO5 reconstruction as for SLS is performed.

The PLIC normal evaluation is done using standard finite differences on ϕ. Finally, the f field is clipped after each directional advection. However, the current implementation does not modify f based on ϕ. This choice has been made to keep strict mass conservation of the VOF scheme.

The ϕ reset is only performed in mixed cells, including the relaxation proposed by Le Chenadec and Pitsch [36] while the HJ-WENO5 scheme is used otherwise as presented in [218].

The f and ϕ updates are summarized in the following algorithm:

1. Set $f^s = f^n$

2. Compute the compression/dilatation factor f_c using Eq. (3.33) from f^n
3. Perform a PLIC reconstruction by computing $\mathbf{n}$ from ϕ^s and d in all cells were $0 < f^s < 1$
4. Solve Eq. (4.1) and Eq. (4.10) in a given direction to find f^{s+1} and ϕ^{s+1}
5. Clip f^{s+1} using η_c
6. Repeat step 2 to 4 for all directions to obtain f^{n+1} and ϕ^*
7. Perform the redistancing step on ϕ^* to obtain ϕ^{n+1}

The direction order is also alternately swapped at each timestep using cyclic permutation.

4.1.5 Summary of the interface capturing methods

A summary of the four interface capturing methods compared in this chapter is provided in Table 4.1. The timestep has a stability constraint based on the CFL:

$$\Delta t < \frac{\Delta x}{2\|\mathbf{u}\|} \tag{4.11}$$

method	VOF	SLS	ACLS	CLSVOF
Time int.	RK1	RK2	RK3	RK1
$F^{(c)}$	WY	WENO5	BHOUC5	WY-WENO5
Reconstruction	PLIC	–	–	PLIC
Reinit	–	HJ-WENO5	Eq. (3.68)	HJ-relax
$\mathbf{n}$	ELVIRA	Eq. (3.73)	Eq. (3.61)	Eq. (3.73)

Table 4.1: *Summary of computation choices for interface capturing methods*

4.2 Metrics of error

Error metrics have to be introduced in order to evaluate accuracy and conservation properties. This could be based directly on conventional norms applied on c. However, if one wants to compare different interface tracking methods with a different color function c, it is important to define shared metrics that can be comparable. For example, the comparison between f field and ϕ field has to be done by either turning f into a distance function, or ϕ into a volume fraction.
This section aims to detail a shared metric for accuracy assessment. A discussion on the quantification of mass loss is also provided.

4.2.1 Shape errors

The shape metric can be based on the sharpest possible Heaviside H^0 computed from the color function which is defined as

$$H^0_\Gamma(c) = \begin{cases} 1 & \text{if } \phi > 0, \psi > 0.5 \text{ or } f > 0.5 \\ 0 & \text{otherwise} \end{cases} . \tag{4.12}$$

The use of H_Γ^0 results in testing if the cell center $\mathbf{x}_\mathcal{C}$ is in the liquid or the gas phase. An error metric based on $H_\Gamma^0(c)$ can show spurious trends in the convergence rate because of its sharp definition and will not be used for the present study.
To provide an equal base of comparison for accuracy, the shape error E_{shape} is based on the regularized Heaviside function $H_\Gamma^\epsilon(c)$ which is defined as Eq. (2.37) for SLS, ψ for CLS and f for VOF and CLSVOF. They all are smooth versions of χ with an interface thickness of $2\epsilon = \Delta x$ such that $H_\Gamma^\epsilon(c) \underset{\Delta x \to 0}{\longrightarrow} \chi$. Note that f is not exactly a regularized function, but it does not impact the relevance of the comparison as it also has a width of ϵ. The accuracy error is then defined as

$$E_{shape} = \sum_{i=1}^{N_\mathcal{C}} |H_\Gamma^\epsilon(c_{i,T}) - H_\Gamma^\epsilon(c_{i,0})| \mathcal{V}_i \quad , \tag{4.13}$$

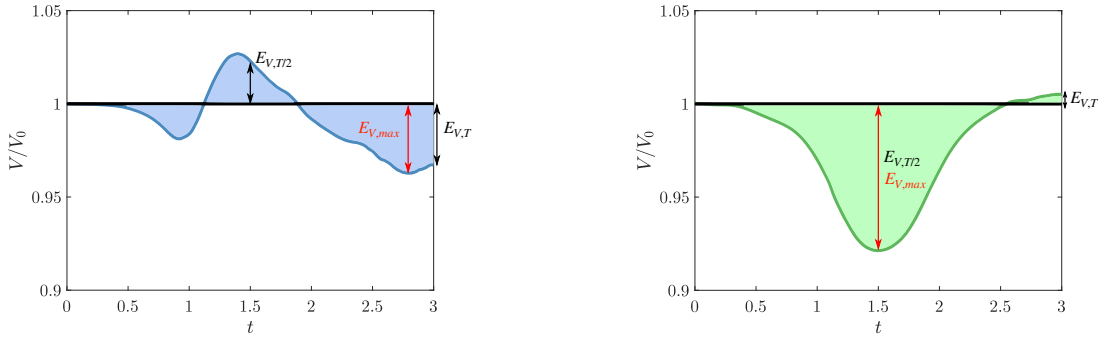
with $t = 0$ the initial time of the simulation, $t = T$ the final time, $N_\mathcal{C}$ the number of cells in the computational domain and $\mathcal{V}_i$ the volume of the cell $\mathcal{C}_i$.

4.2.2 Mass error

Regarding mass conservation, VOF and CLSVOF achieve it at machine precision and will not be displayed. The ACLS method conserves ψ up to machine precision but does not correspond exactly to the volume enclosed in the 0.5 isocontour. This is why a simplex decomposition is performed in each cell to find the intersections between the cell and the interface and compute the related volume. This method leads to a second-order approximation of a volume enclosed in a given isocontour (more details are provided in Section 8.1.3. This same approach is performed for enclosed volume in the zero-isocontour of SLS.
In the literature, the mass error is often defined as the difference between initial and final enclosed volume V_0 and V_T

$$E_{V,T} = \frac{|V_0 - V_T|}{V_0} \quad . \tag{4.14}$$

This metric only gives insights into the change of mass at the end of the simulation, which can hide compensation processes. To illustrate this, temporal evolution of the mass is displayed in Fig. 4.1 for one of the following test cases. While SLS has spurious mass loss oscillating around the V_0 value, the ACLS method provides a symmetric behaviour with respect to time.
If one chooses the standard error $E_{V,T}$, the ACLS is highly superior to SLS as the mass lost between 0 and $T/2$ is retrieved at T.
However, if one chooses another arbitrary time such as $T/2$ to compare the mass error, then $E_{V,T/2}$ is lower for SLS than ACLS, and the conclusion is totally different based on this metric.
Another choice could be to compare the maximum error of mass during the simulation $E_{V,max}$. This metric also gives better mass conservation for SLS compared to ACLS.
In the current study, the choice is to compare the integrated loss of mass over time. This metric gives more details on the mass variation to avoid misleading conclusions on the



(a) *SLS* **(b)** *ACLS*

Figure 4.1: *Temporal evolution of mass for a sphere deformation problem*

conservation property of the considered numerical schemes.
The mass error is then defined as

$$E_{mass} = \frac{1}{V_0 T} \int_0^T |\Delta V| dt \quad , \tag{4.15}$$

with $\Delta V = V(t + dt) - V(t)$ the variation of liquid volume computed from the simplex decomposition method evaluated at time t and $t + dt$.

4.3 Numerical results

Dynamic test cases are built to transport an interface until it retrieves its initial shape after a period T. Here, VOF, SLS, ACLS and CLSVOF are compared with respect to the two main features of an interface capturing method: transport accuracy and mass conservation based on the metric introduced before. A quick discussion on the computational time is also included.

4.3.1 Zalesak's disk rotation

The Zalesak's disk [234] test case consists in a notched circle of radius 0.15 initially centered at $(0.5, 0.75)$ in a $[1 \times 1]$ domain. The notched width is 0.05 and notched length is 0.25. The velocity field is a solid rotation defined as

$$\mathbf{u} = \begin{pmatrix} 2\pi(0.5 - y) \\ 2\pi(x - 0.5) \end{pmatrix} \quad . \tag{4.16}$$

The results are given for a full rotation of the disk corresponding to a simulation time $T = 1$ for a CFL number of 0.5.

The final shape is compared with the initial condition for all methods in Fig. 4.2. At the lowest resolution 32^2, VOF keeps the notch while SLS is shifted. ACLS and CLSVOF merge

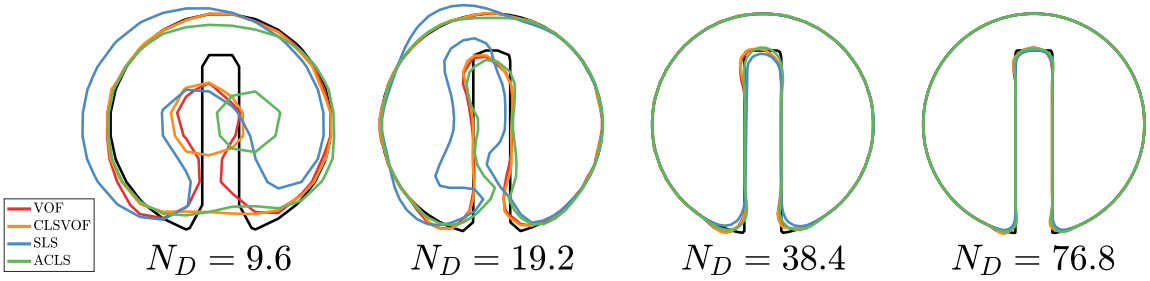
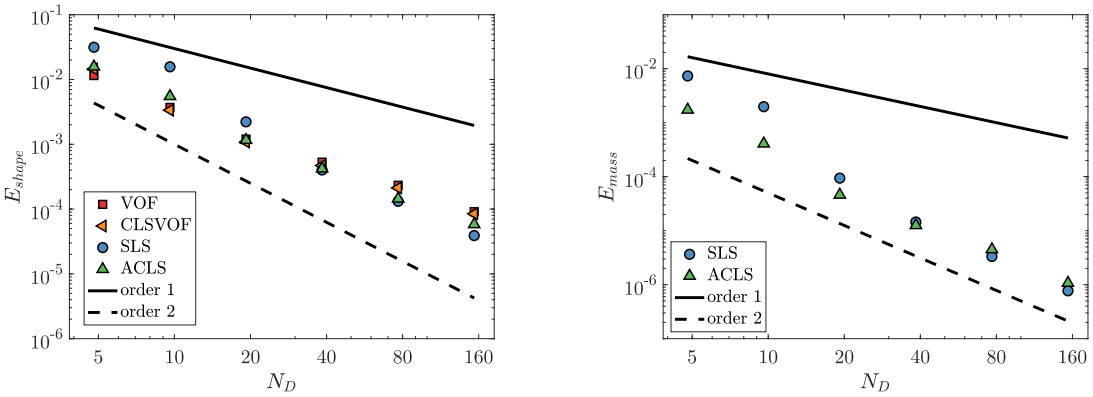


Figure 4.2: *Initial and final shape for the Zalesak's disk rotation*

the two sides of the notch. This shows the difference of normal computation between VOF and CLSVOF: while ELVIRA is able to capture poorly-resolved structures, normals from ϕ tend to merge fronts. From 64^2 resolution, all methods maintain the notch during the whole computation.



(a) E_{shape} *error* **(b)** E_{mass} *error*

Figure 4.3: *Mesh error convergence for the zalesak's disk rotation*

In Fig. 4.3a, the error convergence is displayed for all methods. One can notice that VOF, ACLS and CLSVOF perform well even at meagre resolution, while SLS and ACLS perform better for high resolution with an asymptotic second-order behaviour. Regarding mass conservation, ACLS is better than SLS for the low-resolution meshes while they both have the same conservation properties for the highest resolution. As pictured in Fig. 4.3b.

4.3.2 Vortex in a box

Another classical test case is the vortex-in-a-box first used by Leveque to evaluate high-order advection schemes in incompressibles flows [111]. A circle of radius 0.15 is initially centered at $(0.5, 0.75)$ in a $[1 \times 1]$ domain. The velocity field is deduced from the stream function $\Psi = \frac{1}{\pi}\sin^2(\pi x)\sin^2(\pi y)\cos\left(\pi\frac{t}{T}\right)$ such that it is reversed at $t = T/2$. The results are given for the final time $T = 8$ for an initial CFL number of 0.32 (Δt is constant for the

whole simulation).

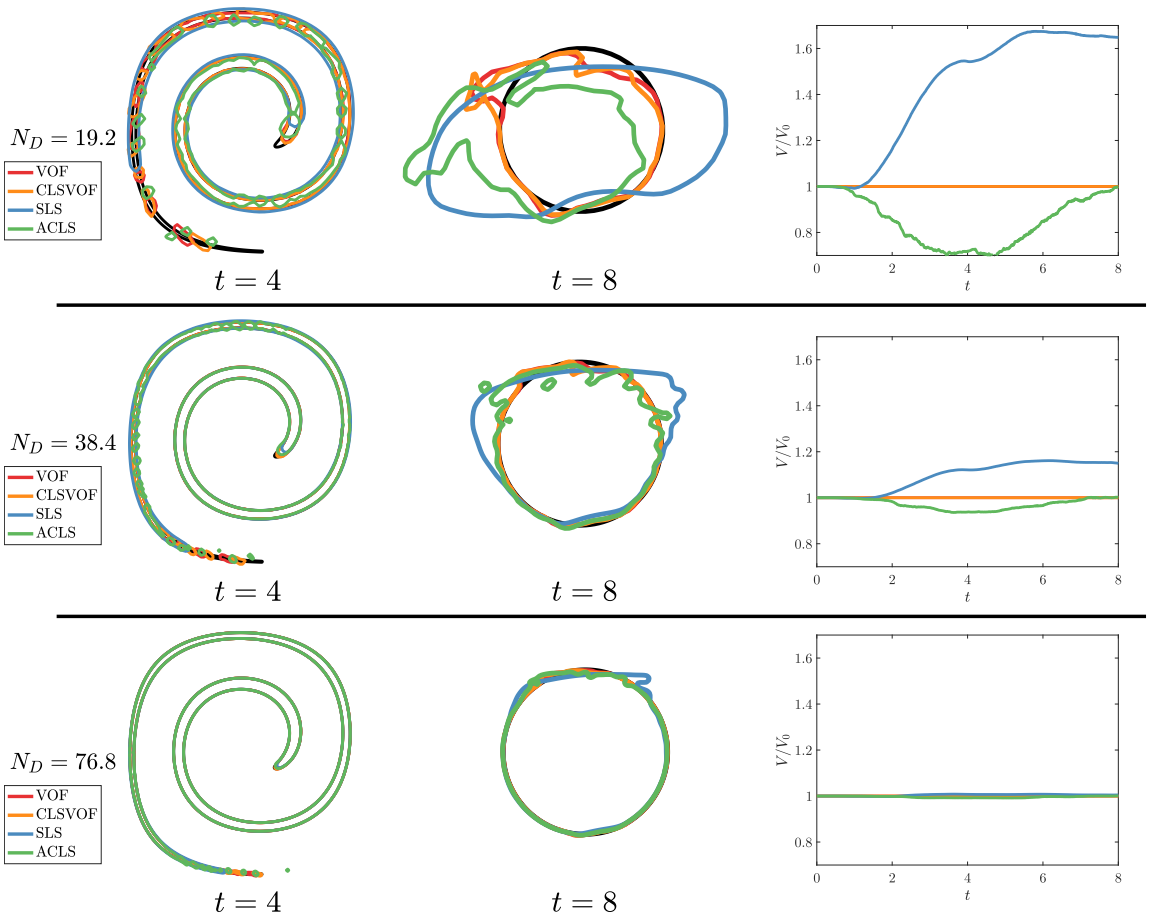
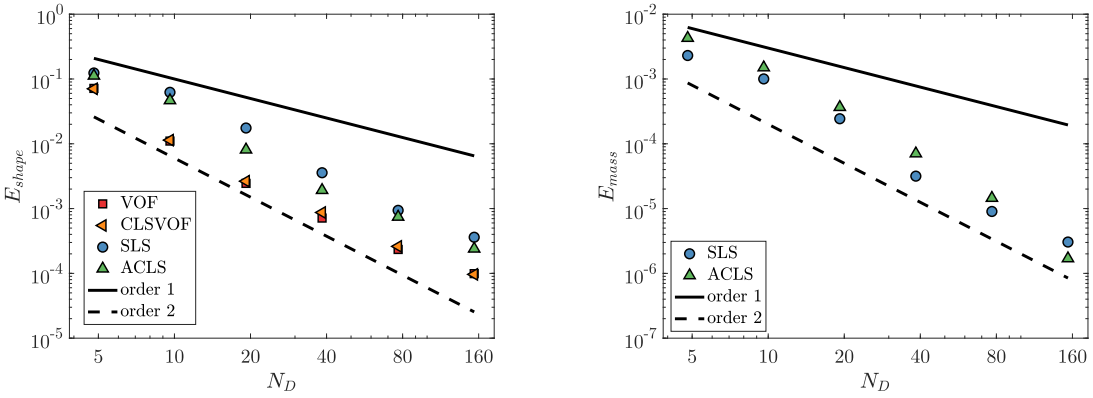


Figure 4.4: *Vortex in a box shape at $t = 4$ and $t = 8$ with the temporal mass evolution*

In Fig. 4.4, VOF, CLSVOF, and ACLS tend to produce numerical atomization in the thinner structures of the serpentine, while the SLS shows a more robust behaviour at the cost of mass conservation. This numerical atomization is less predominant with mesh refinement. In Fig. 4.5a, shape and mass errors are displayed as a function of initial disk resolution. VOF and CLSVOF are performing better for all resolutions. This is expected as the $N_d = 153.6$ case still implies a thin tail, which is not well-resolved. It is interesting to notice that the CLSVOF does not improve the accuracy of the method significantly compared to VOF. Surprisingly, SLS is better at conserving mass than ACLS based on our total volume variation metric. However, the ACLS method is able to retrieve a final mass close to the initial one, which is not the case for SLS.



(a) E_{shape} *error* **(b)** E_{mass} *error*

Figure 4.5: *Mesh error convergence for the vortex in a box*

4.3.3 Sphere deformation

A 3D test case is the sphere deformation, also presented in [111]. A sphere of radius 0.15 is initially centered at $(0.35, 0.35, 0.35)$ in a $[1 \times 1 \times 1]$ domain. It is then advected by a velocity field which induces a combination of stretching in the x-y plane and the x-z plane with an inversion at $t = T/2$.

$$\begin{aligned} u_x &= 2\sin^2(\pi x)\sin(\pi y)\sin(\pi z)\cos\left(\pi\frac{t}{T}\right) \\ u_y &= -\sin(\pi x)\sin^2(\pi y)\sin(\pi z)\cos\left(\pi\frac{t}{T}\right) \\ u_z &= -\sin(\pi x)\sin(\pi y)\sin^2(\pi z)\cos\left(\pi\frac{t}{T}\right) \end{aligned} \quad . \tag{4.17}$$

The results are given for the final time $T = 3$ for an initial CFL number of 0.32 (Δt is kept constant for the whole simulation).

The same conclusions as for the vortex-in-a-box can be drawn from this 3D test case: VOF, CLSVOF and ACLS produce some numerical atomization when the interface is under-resolved, as shown in Fig. 4.6 for $t = 1.5$ at a low mesh resolution of $N_D = 9.6$. This numerical atomization disappears with mesh refinement. ACLS and SLS seem to lose a lot of mass for $N_D = 9.6$ even if ACLS is able to earn back the mass it has lost during the reversed part of the simulation. All methods exhibit a thin tail on the sphere at $t = T$ representing less and less mass with mesh refinement even if it is still present for SLS and ACLS at $N_D = 38.4$. CLSVOF seems to handle this behaviour in the best way for the smallest resolutions.
From Fig. 4.7a, VOF and CLSVOF are still the most accurate methods while ACLS and SLS show similar mass conservation in Fig. 4.7b. Apparently, the transition from 2D to 3D does not affect the overall behaviour of the methods. One slight improvement can be

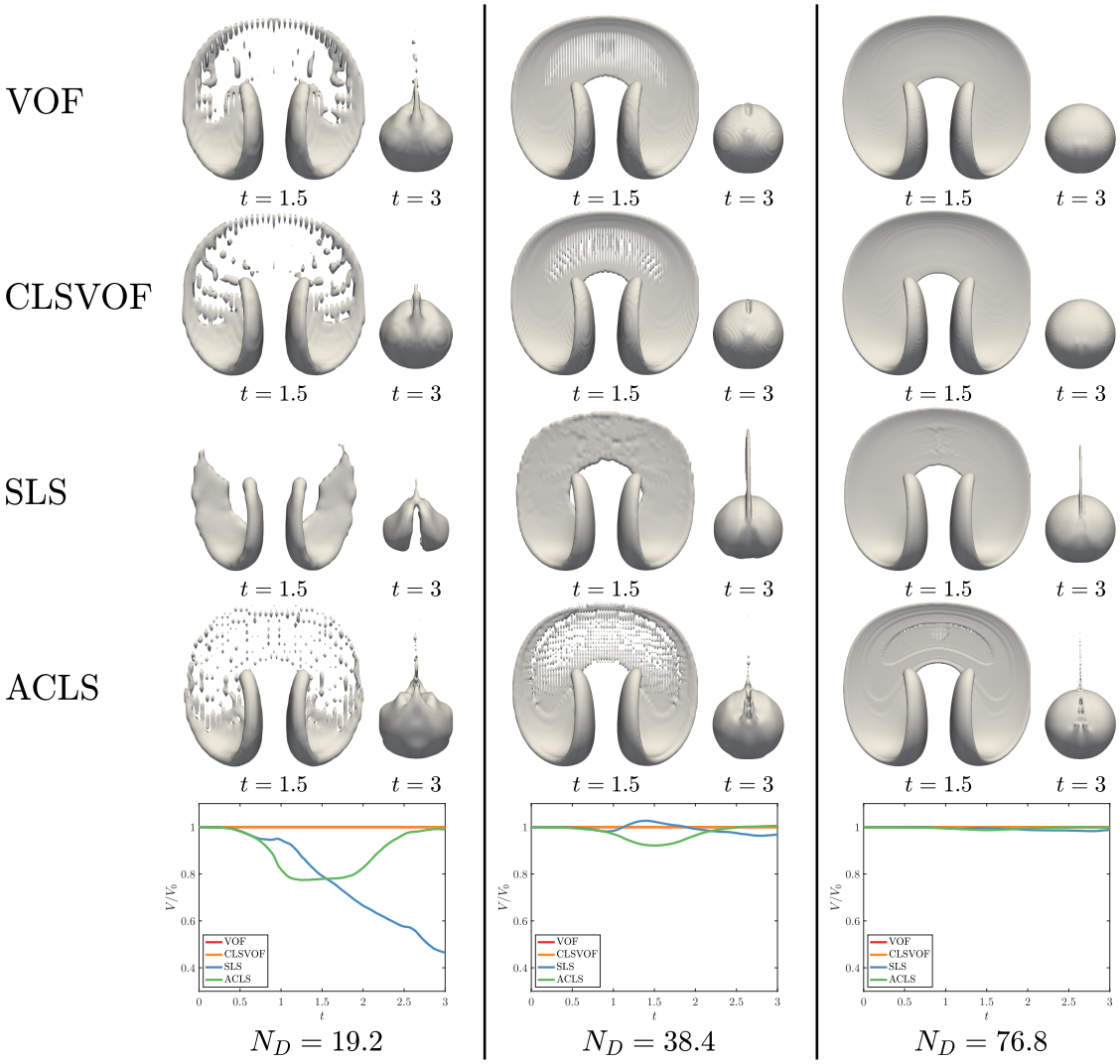


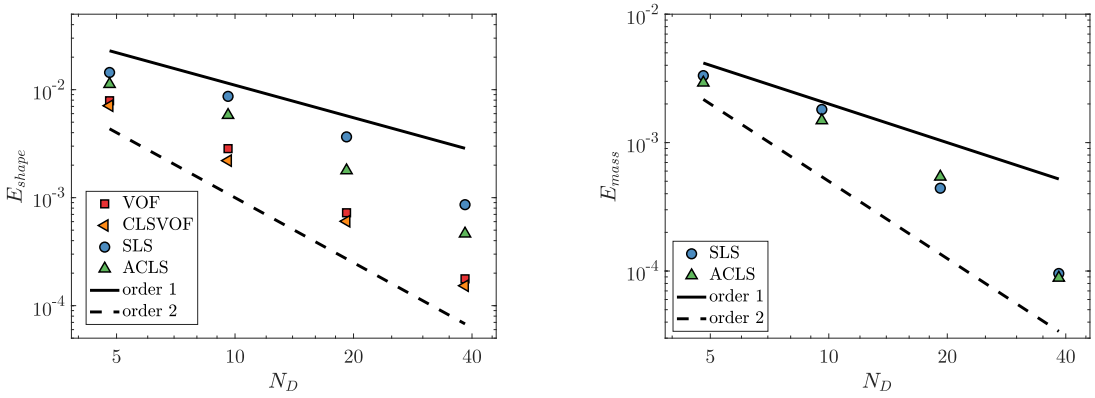
Figure 4.6: *Shape at $t = 1.5$ and $t = 3$ for the sphere deformation*

noticed by using CLSVOF for normal computation in 3D. This can be explained by the ELVIRA accuracy falling behind in 3D configuration if a compact stencil of $3 \times 3 \times 3$ is used. It has been shown that second-order accurate normal computation is only achieved with a stencil of $5 \times 5 \times 5$ [133].

4.3.4 Computational time

To complete the comparison, the computational cost is compared between the methods. In Fig. 4.8, the Reduced Computation Time (RCT) is given

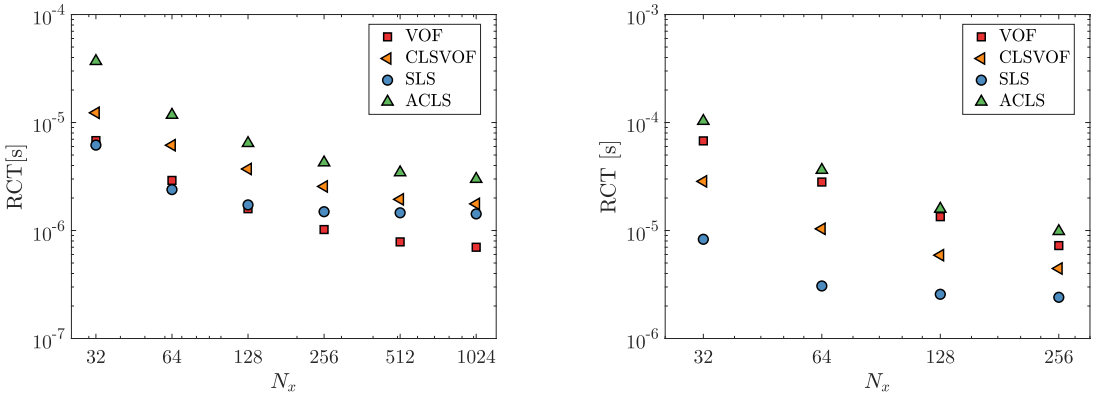
$$\mathrm{RCT} = \frac{\mathrm{WCT} \times N_{CPU}}{N_{\mathcal{C}} \times N_{ite}} \quad , \tag{4.18}$$



(a) E_{shape} *error* **(b)** E_{mass} *error*

Figure 4.7: *Mesh error convergence for the sphere deformation*

with $N_{\mathcal{C}}$ the number of cell, N_{CPU} the number of cores, WCT the Wall Clock Time and N_{ite} the number of iterations .
The 2D vortex in a box case was run on 16 cores, while the 3D deformation case was run on 64 cores. Cores used in this work are Intel Xeon Gold 6230 20C 2.1GHz.



(a) *Vortex in a box* **(b)** *Sphere deformation*

Figure 4.8: *RCT for 2D and 3D cases*

Here, the RCT decreases as the number of elements per cores increases. This is expected as a very coarse mesh, loses a more important part of the simulation time in communications. The cases with more elements show an asymptotic behaviour with a constant RCT.
In 2D configurations, the VOF method is more efficient than CLSVOF, SLS and ACLS and seems to scale better with the number of elements. This is because VOF method only requires to compute fluxes and reconstruction on the interface cells and their neighbours. Hence the computational time does not scale in N_{elem} but in N_Γ. Also, 2D computation of geometric flux and PLIC reconstruction is fast in a split fashion. As expected, SLS is more

efficient than ACLS because of the reinitialization, which is more demanding in the case of ACLS. The CLSVOF method costs approximately VOF and SLS combined.
In 3D, geometry operations are more expensive, and VOF falls behind CLSVOF and SLS regarding efficiency. CLSVOF is more effective than VOF because the normal computation is far less expensive than ELVIRA in 3D.
It is important to remind that these conclusions only hold for the present solver where the same optimization efforts have been given to implement the fourth compared methods.

4.4 Conclusion

A comparison of four popular methods of the literature has been presented focusing on mass conservation and geometrical accuracy for dynamic test cases with imposed velocity. While VOF and CLSVOF are exactly mass conservative, SLS provides a smoother representation of the interface. ACLS shows minor improvements in the mass compared to SLS. Overall, Coupling VOF with LS seems to be the most promising choice in a cartesian finite-volume framework as the conservation properties of VOF are preserved interface representation is more accurate. The versatility of such strategy, taking advantage of the strengths of each method, is of particular interest when additional physics needs to be modelled, such as surface tension or phase-change.
The next part investigates the impact of the interface capturing choice on the simulation of incompressible two-phase flows. In such a case, the color function c is used to close some terms in the resolution of the two-phase flow equations derived in Chapter 2. The consequence is that the conservation and accuracy properties of the interface capturing method directly impacts the accuracy of the velocity field.

Part III

Simulation of two-phase flows without phase change

Chapter 5

Numerical methods for incompressible two-phase flows

Contents

In addition to the interface capturing, two-phase flow simulations are also very challenging because of the interface discontinuities, which must be treated carefully. This chapter aims to provide an overview of the numerical challenges associated with the treatment of interface discontinuities in incompressible two-phase flow solvers without phase change, which are:

1. The evaluation of the curvature κ, which is crucial for a robust and accurate surface tension modelling.

2. The density discontinuity treatment, which also implies momentum discontinuity at the interface. A special treatment for the momentum transport is then required to have a robust and consistent solver for high-density ratios.
3. The viscosity discontinuity, which also needs to be considered in the momentum equation to correctly model the dissipation of kinetic energy close to the interface.

As the literature is mature on the subject, an extended review of these three challenges is provided to motivate the choices made in this work.
In Section 5.1, the numerical framework of the two-phase flow solver is presented with the definition of temporal and spatial operators. Then, Section 5.2 investigates the issues arising from inconsistencies between mass and momentum transport with different techniques employed to solve it. Section 5.3 focuses on the correct discretization of the viscosity operator. As the timestep restriction coming from the diffusion is more restrictive than the one related to the velocity in a majority of the configurations considered here, implicit schemes feasibility is also discussed. Finally, emphasis is given on the surface tension modelling with a complete study of the curvature computation using different interface representations in Section 5.4.

5.1 Numerical framework: The projection method

This section aims to describe the main numerical tools used to solve Eq. (1.35) and Eq. (1.6). It encompasses temporal and spatial operators with special treatments at the interface to handle discontinuities.
First introduced by Chorin [39] for solving the Navier-Stokes equations, the projection method is a class of fractional step methods where the pressure is interpreted as a projection operator which forces a vector field to be divergence-free. The projection method is then divided into two main steps:

- A prediction step to advance the velocity field $\mathbf{u}^n$ in time without taking into account Eq. (1.35), this gives an intermediate velocity field $\mathbf{u}^*$
- A correction step to project the intermediate velocity field $\mathbf{u}^*$ to its divergence-free solution $\mathbf{u}^{n+1}$

The prediction step applied to the two-phase incompressible flow Navier-Stokes equation gives

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} + \frac{1}{\rho}\nabla P = \mathcal{L}_{\text{conv}} + \mathcal{L}_{\text{visc}} + \mathcal{L}_{\text{cap}} \quad , \tag{5.1}$$

with P a pressure approximation obtained from a pressure update, $\mathcal{L}_{\text{conv}}$ the convective operator, $\mathcal{L}_{\text{visc}}$ the viscous operator and $\mathcal{L}_{\text{cap}}$ the surface tension operator. Then, the correction step is written as

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\nabla \Pi \quad , \tag{5.2}$$

with Π^{n+1} a potential constructed such that $\nabla \mathbf{u}^{n+1} = 0$ from the variable coefficient Poisson

equation

$$\nabla \cdot \left(\frac{1}{\rho} \nabla \Pi^{n+1} \right) = \frac{1}{\Delta t} \nabla \cdot \mathbf{u}^* \quad . \tag{5.3}$$

The pressure update is then

$$P^{n+1} = P + \mathcal{L}_{pres}(\Pi^{n+1}) \quad , \tag{5.4}$$

with $\mathcal{L}_{\text{pres}}$ an operator relating P and Π.
The choices of operator discretization, intermediate velocity evaluation and boundary conditions treatment lead to either first or second-order accuracy in time for velocity and pressure (see [23] for a complete analysis).
In the solver, we use a class of projection methods presented by Kim and Moin in [100] as "pressure-free methods" (PFM) where $P = 0$. Then the pressure operator of Eq. (5.4) is simply $\mathcal{L}_{\text{pres}} = \Pi^{n+1}$ and the update is written as $P^{n+1} = \Pi^{n+1}$. This implies that the pressure in our simulations is just a relative pressure that cannot be used directly to a thermodynamic pressure. One advantage of such an approach is the prohibition of pressure gradient error in the momentum equation.
A Marker-and-Cell grid arrangement [77] is used: the pressure is defined at the center of a cell while the velocity is located at the boundaries of the cell. In a two-phase flow, one can consider that all scalars are collocated (located at the center of the cell) while the velocity is staggered (located at the boundaries of the cell) as shown in Fig.5.1 for a 2D control volume.
This grid arrangement allows definition of generic operators for staggered and collocated variables. Moreover, the discrete divergence operator applied to velocity is inherently equal to zero (or, more precisely, equal to the threshold of the elliptic solver used for Eq. (5.3)).
The general staggered operators and variables are mentioned with the subscript $f = u, v$ or w depending on their associated control volume. General collocated operators and variables are mentioned with the subscript c. This gives the following generic operators:

- A staggered gradient operator ∇_f for collocated variables Φ_c, the x-component ∇_u^x is defined as

$$\nabla_u^x \Phi_c|_{i-\frac{1}{2},j} = \frac{\Phi_{c,i,j} - \Phi_{c,i-1,j}}{\Delta x} \quad . \tag{5.5}$$

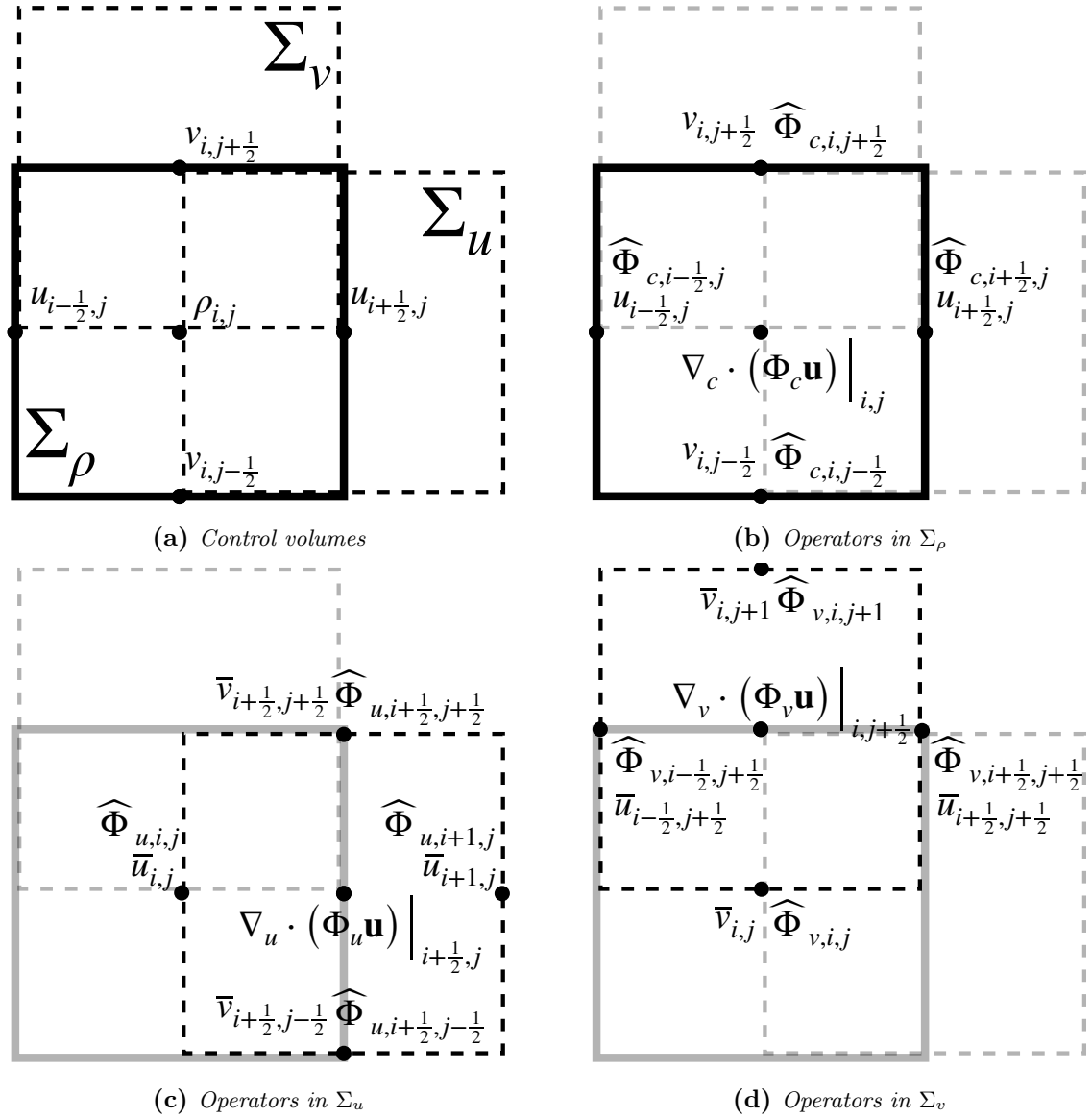
- A collocated gradient operator ∇_c for staggered variable is also introduced with the x-component ∇_c^x of a x-staggered variable defined as

$$\nabla_c^x \Phi_f|_{i,j} = \frac{\Phi_{f,i+\frac{1}{2},j} - \Phi_{f,i-\frac{1}{2},j}}{\Delta x} \quad , \tag{5.6}$$

  with Φ_f a staggered scalar.

- A divergence operator $\nabla_c \cdot (\Phi_c \mathbf{u})$ applied on a control volume Σ_ρ as in Fig. 5.1b

$$\begin{aligned} \nabla_c \cdot (\Phi_c \mathbf{u})|_{i,j} = & \frac{\widehat{\Phi}_{c,i+\frac{1}{2},j} u_{i+\frac{1}{2},j} - \widehat{\Phi}_{c,i-\frac{1}{2},j} u_{i-\frac{1}{2},j}}{\Delta x} \\ & + \frac{\widehat{\Phi}_{c,i,j+\frac{1}{2}} v_{i,j+\frac{1}{2}} - \widehat{\Phi}_{c,i,j-\frac{1}{2}} v_{i,j-\frac{1}{2}}}{\Delta y} \quad , \end{aligned} \tag{5.7}$$



(a) *Control volumes* **(b)** *Operators in* Σ_ρ

(c) *Operators in* Σ_u **(d)** *Operators in* Σ_v

Figure 5.1: *Illustration of the MAC grid arrangement with the associated control volumes and operators. The plain line is the collocated control volume* Σ_ρ *while the dashed lines are the staggered control volumes* Σ_u *and* Σ_v*. Variables* $\widehat{\Phi}$ *are interpolated with arbitrary order at the faces while* $\overline{\mathbf{u}}$ *are linearly interpolated at the faces.*

with $\widehat{\Phi}_c$ an arbitrary interpolation of Φ_c to a face of the Σ_ρ control volume.

- A divergence operator $\nabla_f \cdot (\Phi_f \mathbf{u})$ on a staggered volume control (for Σ_u in Fig. 5.1c)

$$\begin{aligned}\nabla_u \cdot (\Phi_u \mathbf{u})|_{i-\frac{1}{2},j} = & \frac{\widehat{\Phi}_{u,i,j}\overline{u}_{i,j} - \widehat{\Phi}_{u,i-1,j}\overline{u}_{i-1,j}}{\Delta x} \\ & + \frac{\widehat{\Phi}_{u,i-\frac{1}{2},j+\frac{1}{2}}\overline{v}_{i-\frac{1}{2},j+\frac{1}{2}} - \widehat{\Phi}_{u,i-\frac{1}{2},j-\frac{1}{2}}\overline{v}_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y} \quad , \end{aligned} \tag{5.8}$$

  with $\widehat{\Phi}_u$ an arbitrary interpolation of Φ_u and $\overline{u}$ a linear interpolation of velocity to a face of the Σ_u control volume.

The divergence operator is built such that $\nabla_c \cdot \mathbf{u} = 0$ discretely through the projection method. It is also possible to show that $\nabla_f \cdot \mathbf{u} = 0$ discretely as demonstrated by Griffith in [71]

$$\begin{aligned}\nabla_f \cdot \mathbf{u}|_{i-\frac{1}{2},j} &= \frac{\overline{u}_{i,j} - \overline{u}_{i-1,j}}{\Delta x} + \frac{\overline{v}_{i-\frac{1}{2},j+\frac{1}{2}} - \overline{v}_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y} \\ &= \frac{1}{2}\left(\nabla_c \cdot \mathbf{u}|_{i,j} + \nabla_c \cdot \mathbf{u}|_{i+1,j}\right) \qquad = 0 \quad . \end{aligned} \tag{5.9}$$

In the following, if a quantity is located at its natural definition in a MAC grid arrangement, the subscript is discarded for clarity. For example, $\mathbf{u}$ is the staggered velocity with u defined in Σ_u, and P is the collocated pressure.

5.2 Momentum transport

Previously, mass conservation has been investigated for the different interface capturing approaches. It is also important to introduce the notion of momentum conservation which is not straightforward in a two-phase flow solver.
Rewriting Eq. (5.1) without considering viscous, gravity and surface tension contribution gives

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\mathbf{u}^n \cdot \nabla \mathbf{u}^n \quad . \tag{5.10}$$

Equation (5.10) applied to the first velocity component u in the control volume Σ_u gives

$$\frac{u^* - u^n}{\Delta t} = -\nabla_u \cdot (u\mathbf{u}) \quad , \tag{5.11}$$

with $\nabla_u \cdot$ the staggered divergence operator described by Eq. (5.8). The prediction step Eq. (5.1) directly derives from a one-phase incompressible flow formulation where both velocity and momentum are continuous across the interface. This equation is written in velocity form where the transported variable is $\mathbf{u}$. Raessi et al. argued in [176] that using this form allows a simple treatment of the non-linear convection operator $\mathcal{L}_{\text{conv}} = -\mathbf{u}^n \cdot \nabla_f \mathbf{u}^n$ as $\mathbf{u}$ is continuous across the interface even for a two-phase flow. The transported velocity $\widehat{u}$ need to be interpolated to the faces of Σ_u. One needs to be careful about the Total Variation Diminishing (TVD) and Total Variation Bounded (TVB) behaviours of the

interpolation to avoid wiggles on the velocity field. The most popular interpolations are using high-order upwind schemes such as ENO/WENO [93] or QUICK [109] which have low dissipation and limited dispersion.
This method is particularly straightforward to implement and well-suited for low density ratios. However, using a velocity form implies a lack of consistency between mass fluxes $F^{(\rho)}$ and momentum fluxes $F^{(\rho\mathbf{u})}$ as only $F^{(\mathbf{u})}$ fluxes are explicitly computed. It can lead to huge errors in momentum conservation for high-density ratios.
A solution is to consider the conservative form Eq. (1.6) where the transported variable is the momentum $\rho\mathbf{u}$:

$$\frac{\rho_f^{n+1}\mathbf{u}^* - \rho_f^n\mathbf{u}^n}{\Delta t} = -\nabla\cdot(\rho\mathbf{u}\mathbf{u}) \quad . \tag{5.12}$$

By rearranging the terms, the resulting time integration of the primitive variable $\mathbf{u}$ can be written as

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\frac{1}{\rho_f^{n+1}}\nabla\cdot(\rho\mathbf{u}\mathbf{u}) - \frac{\mathbf{u}^n}{\rho^{n+1_f}}\frac{\rho_f^{n+1} - \rho_f^n}{\Delta t} \quad . \tag{5.13}$$

The update of $\mathbf{u}^*$ is now a velocity transport with an additional term corresponding to a flux correction.
The resolution of Eq. (5.13) leads to three numerical issues:

(1) There is a discontinuity of momentum at the interface because of the density jump. The transport of $\rho\mathbf{u}$ is more challenging than $\mathbf{u}$ and requires special treatments near the discontinuity. The extension to high-order schemes is then non-trivial to apply in this context.
(2) The MAC grid arrangement gives a different control volume definition for each variable. In Fig. 5.1, a mass control volume Σ_ρ is represented in plain line while momentum control volumes in Σ_u and Σ_v direction are represented in dashed lines. respectively. The consistency between mass fluxes and momentum fluxes is then hard to achieve.
(3) In a VOF context, fluxes of volume fraction $F^{(f)}$ are directly related to fluxes of mass $F^{(\rho)}$ through the relation

$$F^{(\rho)} = \rho_l F^{(f)} + \rho_g F^{(1-f)} \quad . \tag{5.14}$$

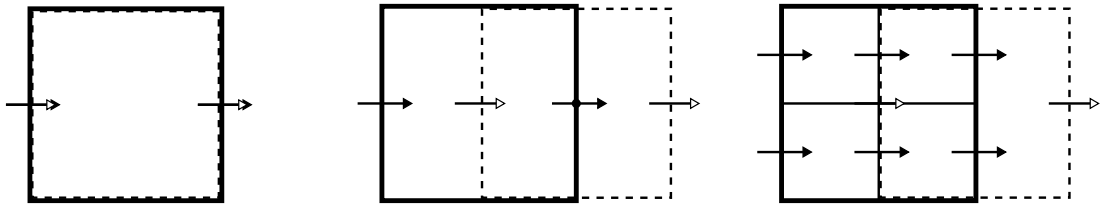
However, Level-Set does not provide such a direct link between $F^{(\phi)}$ and $F^{(\rho)}$ from the equation derivation as discussed in Chapter 2. It explains why Level-Set fails at conserving strictly mass and, by extension, momentum.

A fully consistent method needs to reach the following requirements:

(i) The mass fluxes have to be discretely reused in the momentum fluxes.
(ii) The staggered mass fluxes have to be determined from the collocated mass fluxes used to update c.

As described in [64], this requirement (i) is met if the momentum fluxes are computed (here for u) as

$$F^{(\rho u)} = uF^{(\rho)} \quad . \tag{5.15}$$



(a) *Collocated approach* **(b)** *Staggered approach* **(c)** *Dual grid approach*

Figure 5.2: *Different choice of control volume definition for consistent mass and momentum configuration. The plain line represent the collocated control volume Σ_ρ while the dashed line is the momentum control volume Σ_u. The black arrows represents the fluxes in the collocated fluxes $F^{(\rho)}$ while the white arrows are the staggered fluxes $F_u^{(\rho)}$.*

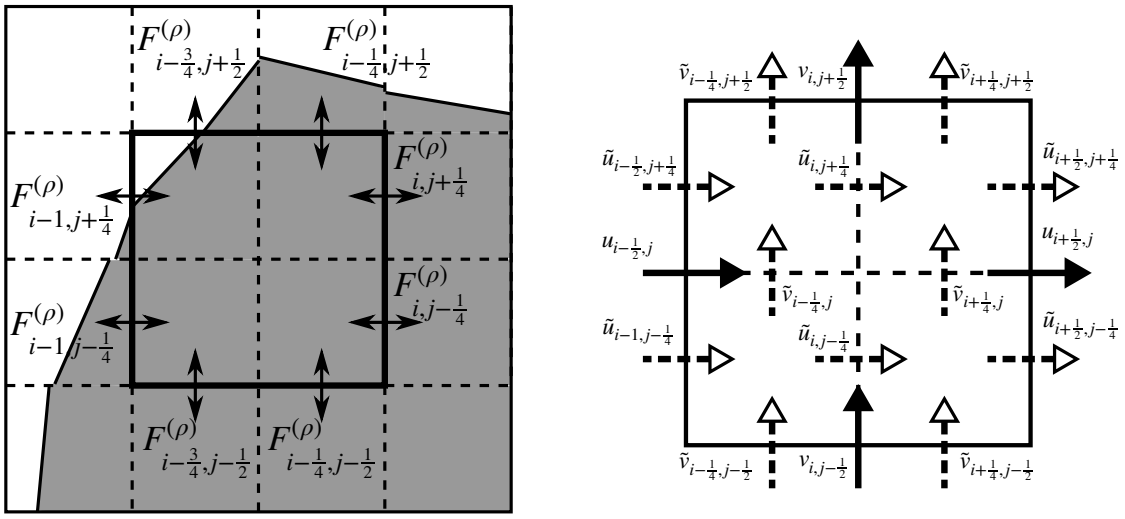
This relation is sufficient to provide consistency between mass and momentum for a given momentum component and solve the robustness issues.
However, requirement (ii) is mandatory for discrete momentum conservation. Achieving (ii) is non trivial and only possible in a VOF framework because of issue (3).
Another difficulty arises from issue (2). Velocity components need fluxes defined in the staggered cells $F_f^{(\rho)}$ while the only mass fluxes available are defined for collocated cells $F_c^{(\rho)}$. The challenge is then to establish a relation between $F_c^{(\rho)}$ and $F_f^{(\rho)}$. There are several ways to compute the staggered fluxes, but they are limited to a VOF framework in the literature [182; 224]. These methods are presented hereafter.

5.2.1 Dual grid

To overcome the problem of grid arrangement, Rudman [182] proposed to use two different grids in a VOF framework to meet requirements (i) and (ii). One primal grid for the momentum solver (plain line on Fig. 5.2c) where pressure is collocated and velocity is staggered, and one dual grid where f is transported (dashed on Fig. 5.2c) which is two times finer. The main advantage of this method is the natural relation between collocated mass fluxes $F_c^{(\rho)}$ of the dual grid and the staggered fluxes of the primal grid $F_f^{(\rho)}$. Indeed, the fluxes are defined in both collocated and staggered control volumes of the primal grid. For example, the mass fluxes associated to the control volume $\Sigma_{u,i-\frac{1}{2},j}$ are pictured on Fig. 5.3a.
Let us consider a mass balance between time t^n and t^{n+1} in the control volume $\Sigma_{u,i-\frac{1}{2},j}$

$$\begin{aligned}\frac{\rho^{n+1}_{u,i-\frac{1}{2},j} - \rho^{n}_{u,i-\frac{1}{2},j}}{\Delta t} = &- \frac{1}{2\Delta x}\left(F^{(\rho)}_{i,j-\frac{1}{4}} + F^{(\rho)}_{i,j+\frac{1}{4}} - F^{(\rho)}_{i-1,j-\frac{1}{4}} - F^{(\rho)}_{i-1,j+\frac{1}{4}}\right) \\ &- \frac{1}{2\Delta y}\left(F^{(\rho)}_{i-\frac{3}{4},j+\frac{1}{2}} + F^{(\rho)}_{i-\frac{1}{4},j+\frac{1}{2}} - F^{(\rho)}_{i-\frac{3}{4},j-\frac{1}{2}} - F^{(\rho)}_{i-\frac{1}{4},j-\frac{1}{2}}\right) \quad .\end{aligned} \tag{5.16}$$

All the above fluxes are available from the dual grid VOF flux computation.
However, using a dual grid requires interpolating velocity on the dual grid and ensuring the



(a) *Fluxes computation in* $\Sigma_{u,i-\frac{1}{2},j}$ **(b)** *Velocity definition in* $\Sigma_{\rho,i,j}$

Figure 5.3: *Dual grid fluxes and velocity definition. The plain line represent the* $\Sigma_{u,i-\frac{1}{2},j}$ *control volume while the dashed lines represent the dual grid where* f *is transported. The double arrows of Fig. 5.3a are the mass fluxes. The plain and dashed arrows of Fig. 5.3a represents the velocity of the primal and dual grid respectively.*

divergence-free properties of the interpolated velocity field for a correct VOF advection. In the original paper [182], a simple interpolation is proposed:

- External dual grid velocities $\tilde{u}_{i-\frac{1}{2},j-\frac{1}{4}}$, $\tilde{u}_{i-\frac{1}{2},j+\frac{1}{4}}$, $\tilde{u}_{i+\frac{1}{2},j-\frac{1}{4}}$, $\tilde{u}_{i+\frac{1}{2},j+\frac{1}{4}}$, $\tilde{v}_{i-\frac{1}{4},j-\frac{1}{2}}$, $\tilde{v}_{i+\frac{1}{4},j-\frac{1}{2}}$, $\tilde{v}_{i-\frac{1}{4},j+\frac{1}{2}}$, $\tilde{v}_{i+\frac{1}{4},j+\frac{1}{2}}$ are roughly taken as the values of their respective primitive cell velocity. For the face at $A_{i-\frac{1}{2},j}$ this leads to

$$\tilde{u}_{i-\frac{1}{2},j-\frac{1}{4}} = \tilde{u}_{i-\frac{1}{2},j+\frac{1}{4}} = u_{i-\frac{1}{2},j} \quad , \tag{5.17}$$

  which is a first-order interpolation of the velocity.
- Internal dual grid velocities $\tilde{u}_{i,j-\frac{1}{4}}$, $\tilde{u}_{i,j+\frac{1}{4}}$, $\tilde{v}_{i-\frac{1}{4},j}$, $\tilde{v}_{i+\frac{1}{4},j}$ are then deduced as a linear interpolation of external dual grid velocities, for $\tilde{u}_{i,j-\frac{1}{4}}$ this gives

$$\tilde{u}_{i,j-\frac{1}{4}} = \frac{1}{2}\left(\tilde{u}_{i-\frac{1}{2},j-\frac{1}{4}} + \tilde{u}_{i+\frac{1}{2},j-\frac{1}{4}}\right) \quad . \tag{5.18}$$

Note that this is still an overall first-order interpolation, as this gives $\tilde{u}_{i,j-\frac{1}{4}} = \tilde{u}_{i,j+\frac{1}{4}}$. It can be shown that this interpolation procedure leads to discrete divergence-free velocity

field $\tilde{\mathbf{u}}$. Let us write the discrete divergence operator for the dual grid cell $\tilde{\mathcal{C}}_{i-\frac{1}{4},j-\frac{1}{4}}$:

$$\begin{aligned}
\nabla\cdot\tilde{\mathbf{u}}|_{i-\frac{1}{4},j-\frac{1}{4}} &= \frac{\tilde{u}_{i,j-\frac{1}{4}}-\tilde{u}_{i-\frac{1}{2},j-\frac{1}{4}}}{\Delta x}+\frac{\tilde{v}_{i-\frac{1}{4},j}-\tilde{v}_{i-\frac{1}{4},j-\frac{1}{2}}}{\Delta y}\\
&= \frac{\frac{1}{2}\left(u_{i+\frac{1}{2},j}+u_{i-\frac{1}{2},j}\right)-u_{i-\frac{1}{2},j}}{\Delta x}+\frac{\frac{1}{2}\left(v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}\right)-v_{i,j-\frac{1}{2}}}{\Delta y}\\
&= \frac{1}{2}\left(\frac{u_{i+\frac{1}{2},j}-u_{i-\frac{1}{2},j}}{\Delta x}+\frac{v_{i,j+\frac{1}{2}}-v_{i,j-\frac{1}{2}}}{\Delta y}\right)\\
&= \frac{1}{2}\,\nabla_c\cdot\mathbf{u}|_{i,j} \quad . \qquad (5.19)
\end{aligned}$$

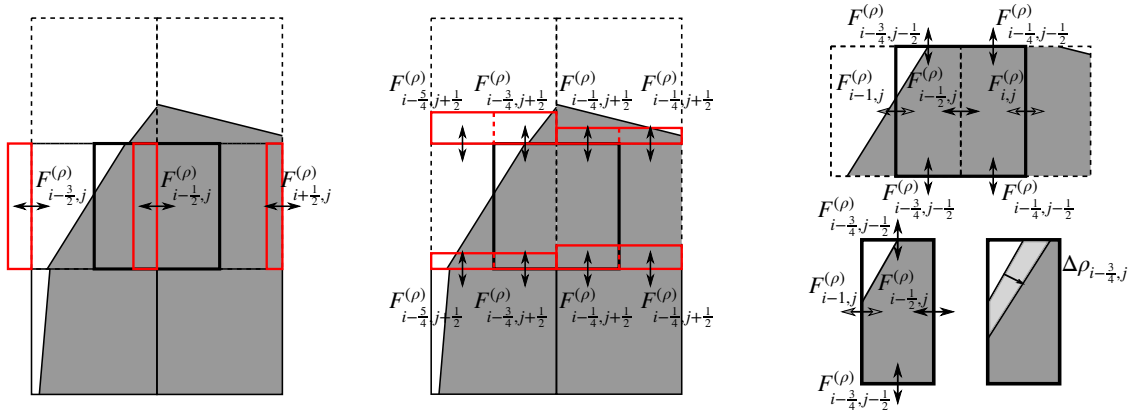
It follows that $\nabla\cdot\tilde{\mathbf{u}}|_{i-\frac{1}{4},j-\frac{1}{4}}=0$ discretely. The same can be done for all the dual grid cells.
Note that higher-order interpolation can be performed to improve the accuracy of the VOF advection on the dual grid (see [221] and [18] for details), but they require to solve a small linear system in all primal cells.
This dual grid approach has also been implemented in an unsplit VOF advection framework by Owkes and Desjardins [158].
This strategy allows a straightforward definition of the staggered fluxes at the cost of a dual grid transport of VOF, leading to a critical computational cost increase in 3D configurations (number of elements multiplied by 8). In fact, using a dual grid is not necessary to find a relation between staggered and collocated fluxes.

5.2.2 Single grid

In [224], Vaudor et al. proposed an alternative to the dual grid approach which is able to respect requirements (i) and (ii). In fact, the PLIC reconstruction on the primal grid already contains all the information required to compute the staggered mass fluxes. By reconstructing the volume fraction at t^n and t^{n+1} on half control volumes based on PLIC, a painstaking mass balance can be performed to retrieve the staggered mass fluxes from half volume fractions and collocated mass fluxes, as illustrated in Fig. 5.4c.
This mass balance between time t^n and t^{n+1} in the control volume $\Sigma_{u,i+\frac{1}{2},j}$ can be written as

$$\begin{aligned}
\frac{\rho^{n+1}_{u,i-\frac{1}{2},j}-\rho^{n}_{u,i-\frac{1}{2},j}}{\Delta t} &= \frac{1}{\Delta x}\left(F^{(\rho)}_{i,j}-F^{(\rho)}_{i-1,j}\right)\\
&\quad+\frac{1}{\Delta y}\left(F^{(\rho)}_{i-\frac{1}{2},j+\frac{1}{2}}-F^{(\rho)}_{i-\frac{1}{2},j-\frac{1}{2}}\right) \quad . \qquad (5.20)
\end{aligned}$$

The VOF fluxes are computed from the PLIC reconstruction as presented in Figs. 5.4a and 5.4b. Fluxes are only available at the control volume faces Σ_ρ using the VOF fluxes, however, it is possible to deduce Σ_u fluxes from them.
The $F^{(\rho)}$ fluxes can be split into two half fluxes without any mass balance violation such



(a) $F^{(\rho)}$ *fluxes in the x-direction*

(b) $F^{(\rho)}$ *fluxes in the y-direction*

(c) $\Delta\rho$ *computation in the half cell*

Figure 5.4: *Fluxes computation for the control volume* $\Sigma_{u,i-\frac{1}{2},j}$. *The plain line represent the* $\Sigma_{u,i-\frac{1}{2},j}$ *control volume while the dashed lines represent the collocated grid where* f *is transported. The double arrows are the mass fluxes* $F^{(\rho)}$. *The plain and dashed arrows of Fig. 5.3a represents the velocity of the primal and dual grid respectively*

that $F^{(\rho)}_{i-1,j-\frac{1}{2}} = F^{(\rho)}_{i-\frac{5}{4},j-\frac{1}{2}} + F^{(\rho)}_{i-\frac{3}{4},j-\frac{1}{2}}$ and $F^{(\rho)}_{i,j-\frac{1}{2}} = F^{(\rho)}_{i-\frac{1}{4},j-\frac{1}{2}} + F^{(\rho)}_{i+\frac{1}{4},j-\frac{1}{2}}$. Hence, the $F^{(\rho)}$ fluxes at $\Sigma_{u,i-\frac{1}{2},j}$ top flux can be computed as

$$F^{(\rho)}_{i-\frac{1}{2},j+\frac{1}{2}} = F^{(\rho)}_{i-\frac{3}{4},j+\frac{1}{2}} + F^{(\rho)}_{i-\frac{1}{4},j+\frac{1}{2}} \quad , \tag{5.21}$$

and the same procedure can be performed to retrieve the bottom flux $F^{(\rho)}_{i-\frac{1}{2},j-\frac{1}{2}}$.

The computation of $F^{(\rho)}$ is more tricky as the fluxes $F^{(\rho)}_{i,j}$ and $F^{(\rho)}_{i-1,j}$ are not directly available. The idea here is to write the mass balance on half of the control volume $\Sigma_{u,i-\frac{3}{4},j}$ as shown in Fig. 5.4c

$$\begin{aligned}\frac{\rho^{n+1}_{u,i-\frac{3}{4},j} - \rho^{n}_{u,i-\frac{3}{4},j}}{\Delta t} &= \frac{1}{\Delta x}\left(F^{(\rho)}_{i-\frac{1}{2},j} - F^{(\rho)}_{i-1,j}\right) \\ &+ \frac{1}{\Delta y}\left(F^{(\rho)}_{i-\frac{3}{4},j+\frac{1}{2}} - F^{(\rho)}_{i-\frac{3}{4},j-\frac{1}{2}}\right) \quad .\end{aligned} \tag{5.22}$$

In this equation, the only unknown is the flux $F^{(\rho)}_{i-1,j}$.

Indeed, the other fluxes are already available and $\frac{\rho^{n+1}_{u,i-\frac{3}{4},j} - \rho^{n}_{u,i-\frac{3}{4},j}}{\Delta t} = \Delta\rho_{i-\frac{3}{4},j}$ is the variation in the half control volume between t^n and t^{n+1} which can be computed easily from the PLIC reconstructions at these times (light grey area of Fig. 5.4c . Equation (5.22) can be then rewritten as

$$F^{(\rho)}_{i-1,j} = F^{(\rho)}_{i-\frac{1}{2},j} + \frac{\Delta x}{\Delta y}\left(F^{(\rho)}_{i-\frac{3}{4},j+\frac{1}{2}} - F^{(\rho)}_{i-\frac{3}{4},j-\frac{1}{2}}\right) - \frac{\Delta x}{\Delta t}\left(\rho^{n+1}_{u,i-\frac{3}{4},j} - \rho^{n}_{u,i-\frac{3}{4},j}\right) \quad . \tag{5.23}$$

This whole procedure can be applied in the same manner for Σ_v or Σ_w control volumes and 3D extension is straightforward.
With this method, requirements (i) and (ii) are met by using only one grid. It is then a very encouraging method to handle high-density ratios and conserve momentum.

5.2.3 Additional continuity equations

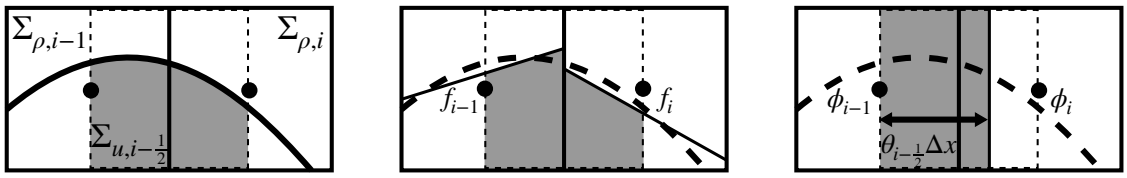
In a Level-Set framework, Moureau & Desjardins [47] proposed to solve an additional continuity equation in each of the momentum control volumes Σ_u, Σ_v (and Σ_w in 3D) in order to retrieve mass fluxes.
The new term in Eq. (5.13) requires to solve a new continuity equation $\frac{\rho_f^{n+1}-\rho_f^n}{\Delta t}$ for each $\mathbf{u}$ component.
First, the staggered density ρ_f^n need to be computed from the interface color function. For example, $\rho^n_{u,i-\frac{1}{2}}$ is computed for the $\Sigma_{u,i-\frac{1}{2}}$ control volume as

$$\rho^n_{u,i-\frac{1}{2}} = \rho_g + \theta_{i-\frac{1}{2}}\,[\rho]_\Gamma \quad , \tag{5.24}$$

with $\theta_{i-\frac{1}{2}}$ a staggered volume fraction as illustrated in Fig. 5.5a In the original paper based on a LS formulation, $\theta_{i-\frac{1}{2}}$ is computed using ϕ^n:

$$\theta_{i-\frac{1}{2}} = \frac{|\phi^n_{i-1}|_+ + |\phi^n_i|_+}{|\phi^n_{i-1}| + |\phi^n_i|} \quad . \tag{5.25}$$

This results in a first-order reconstruction of the volume contained in Σ_u as illustrated in Fig. 5.5c.
In a VOF framework, this value can be retrieved at second-order using the PLIC reconstruction of right half cell $\Sigma_{\rho,i-1}$ and the left half cell of $\Sigma_{\rho,i}$ as shown in Fig. 5.5b. This method has been successfully applied in [159]. A recent method proposed in [238] is using the same approach but the additional continuity equations are updated from geometric fluxes.



(a) *Exact reconstruction* **(b)** *PLIC reconstruction* **(c)** *GFM reconstruction*

Figure 5.5: *Reconstruction of $\rho_{u,i-\frac{1}{2}}$ on a 2D cartesian grid. The grey area represent the fraction $\theta_{i-\frac{1}{2}}$ evaluated in the control volume $\Sigma_{u,i-\frac{1}{2}}$.*

More generally, if a good reconstruction of the volume in a momentum control volume can be achieved, then this method can be applied to ensure a tight coupling between mass and momentum fluxes. It is only true if the same scheme is used to transport ρ_f and $\rho_f \mathbf{u}_f$. The main consequence of using additional continuity equations is the loss of the requirement

(ii). This flux correction avoids the potential discrepancy between mass and momentum when ρ_f^{n+1} is computed as a simple average of ρ^{n+1} computed from ϕ^{n+1}, ψ^{n+1} or f^{n+1} because it meets requirement (i).

5.2.4 Other approaches

5.2.4.1 Collocated fluxes

In the collocated approach, the control volumes Σ_ρ, Σ_u, Σ_v (and Σ_w in 3D) are the same (see Fig. 5.2a). This method was first introduced by Bussmann in a VOF framework [25] where two definitions of the velocity are provided: a collocated velocity $\mathbf{u}_c$ which is not divergence-free discretely and a staggered velocity $\mathbf{u}_f$ which respects the discrete divergence-free condition of Eq. (1.35).
The idea is to transport $\rho_c \mathbf{u}_c$ using Eq. (5.15) with $F^{(c)}$ obtained from the VOF transport. Then a staggered velocity is obtained from interpolation of $\mathbf{u}_c$ and projected into a divergence-free velocity. Then, the staggered velocity is remapped on the collocated points to find the update of $\mathbf{u}_c$. This can be summed up into the following algorithm:

1. Compute $F^{(\rho)}$ from the PLIC reconstruction in a geometric fashion using the staggered velocity $\mathbf{u}_f^n$.
2. Advance the collocated momentum equation to obtain $\mathbf{u}_c^*$ by reusing $F^{(\rho)}$.
3. The predicted staggered velocity $\mathbf{u}_f^*$ is computed by remapping $\mathbf{u}_c^*$.
4. Solve the classical Poisson equation Eq. (5.3) using $\nabla \cdot \mathbf{u}_f^*$.
5. Correct the staggered velocity field from Eq. (5.2) to obtain $\mathbf{u}_f^{n+1}$.
6. Finally, remap $\mathbf{u}_f^{n+1}$ to obtain the updated collocated velocity $\mathbf{u}_c^{n+1}$.

Even if this method was applied first in the VOF framework by using the equivalence between the continuity equation and the VOF transport equation, it is also possible to use such an approach in the Level-Set framework. Ghods & Herrmann proposed a method in [66] where the density at time n is computed from a smooth Heaviside $H_\Gamma^\epsilon(\phi)$.
In previous method $F^{(\rho)}$ is deduced from the VOF fluxes. Hence, ρ_c is totally defined by the evolution equation of f and ρ_c^{n+1} can be directly reused in the next time-step as ρ_c^n. In a Level-Set framework, the continuity equation is explicitly solved in addition to the ϕ transport and reinitialization and the predicted field ρ_c^{n+1} is used to update $\mathbf{u}_c^*$ and discarded hereafter. ρ^{n+1} is then recomputed with $H_\Gamma(\phi^{n+1})$.
The collocated fluxes provide a natural way to transport mass and momentum consistently. However, it requires additional velocity remapping, which can be tricky to handle when surface tension is also included in the momentum equation. In [62; 81], the remapping is treated carefully to avoid numerical issues for surface-tension driven flows. A comparison of momentum-conserving method based on collocated velocity with a non-conserving method can be found in [235]. The momentum fluxes constructed from the VOF fluxes show convergence with a particular increase in accuracy for the coarse meshes when applied to a rising-bubble test case.

5.2.4.2 Jump condition formulation

In this last approach, the momentum discontinuity at the interface is handled by using a liquid velocity and a gas velocity which are defined in their respective phase. The phase velocities are then extrapolated from one phase to the other to have valid values in the discretization of the different terms close to the interface. This method has been proposed by 3 different research teams in 2007: [33] and [218] in the context of phase-change and [215] in a classic two-phase flow solver. The philosophy is to avoid any mixing of liquid and gas velocity by only using phase velocity in the convection operator stencil. If the stencil crosses the interface, then ghost cells obtained from constant extrapolation are used instead of the wrong phase velocities. Then both velocities are coupled through the velocity jump condition prescribed in the Poisson equation Eq. (5.3).
The method can be summed up in the following steps

1. Advance liquid and gas velocities through the prediction step to get $\mathbf{u}_l^*$ and $\mathbf{u}_g^*$
2. Solve the pressure Poisson equation Eq. (5.3) to obtain P with the classic GFM presented in Appendix A.1.
3. Correct each phase velocity in their respective domain and definition to get $\mathbf{u}_l^{n+1}$ and $\mathbf{u}_g^{n+1}$
4. Extend the phase velocities in the other phase suing a methodology of Appendix B.1 to have available ghost values near the interface.

If a velocity node is in the liquid phase, it is then advanced from a liquid prediction step and a liquid pressure correction, while if it is in the gas, it is advanced by the gas prediction step and the gas pressure correction.

5.2.5 Discussion on the discretization limitations

The methods presented above were developed to provide robust schemes for high-density ratios. They all solve the issue of stability by removing velocity spikes at the interface because of inconsistency between mass and momentum transport. However, these methods lead to inherent limitations in both spatial and temporal discretization of the momentum transport. Moreover, all methods which do not rely on the VOF fluxes are not discretely conservative in momentum.

5.2.5.1 Geometric fluxes limitation

First, all methods relying on the VOF fluxes are limited in the temporal integration scheme. In fact, as pointed out in several works [224; 238], the geometric nature of the VOF fluxes is not compatible with most of the higher-order temporal integration schemes such as RK. Then, the momentum equation is only advanced at first-order using a forward Euler step. An alternative proposed in [182] and reused in [224; 238] is to write a predictor-corrector method where a first step is used to find a midpoint velocity at $t^{n+\frac{1}{2}}$ and use it in a second step to advance both mass and momentum. However, this method results in advancing mass and momentum using a forward Euler step with a better approximation of the velocity used in the fluxes. Such a first-order temporal integration can cause some stability issues if one uses high-order flux reconstruction such as WENO5, which is not stable even in linear

stability analysis [138].

5.2.5.2 Splitting fluxes limitation

Another limitation arising from VOF fluxes is their dimensional-splitting nature. This also requires to treat the momentum equation in a dimensional-splitting fashion. If this is not done correctly, then additional temporal splitting errors are introduced in the momentum equation. This issue is discussed in [20] where VOF fluxes are used to transport scalars accordingly. This scheme requires additional flux corrections applied to the transported quantity based on the work of [40].

5.2.5.3 Boundness limitations

Finally, the last limitation discussed in [47] is the boundedness of ρ. As ρ is highly discontinuous at the interface, it is complicated to provide bounded fluxes without decreasing the accuracy of the flux approximation. In fact, in most of the approaches presented above [47; 224; 238], an upwind limiter is used close to the interface (when the stencil of the higher-order reconstruction crosses the interface). Then, all spatial discretizations degenerate to first-order in the narrow band close to the interface.

5.2.5.4 Choice in the solver

In the present solver, the goal is to have a well-suited method for all the interface representations. It means that the momentum transport cannot rely on the VOF fluxes as they are not available when using SLS or ACLS methods. The choice is then to drop the full consistency between the transport of the interface, mass and momentum by supplanted additional continuity equations in the prediction step as proposed in [47]. This method is robust, easy to implement and applicable to SLS and ACLS. Moreover, it allows more flexibility on the flux computation as discussed in Sections 5.2.5.1 and 5.2.5.2. Then, a WENO5 reconstruction can be applied to velocity with an upwind limiter at the interface. The overall method is referred to as WENO5-cons

5.3 Viscosity modelling

When viscosity is discontinuous across the interface ($[\mu]_\Gamma \neq 0$), a jump in the stress tensor arises which can be expressed as a pressure jump $[P]_\Gamma = [2\mu\,(\mathbf{D}\cdot\mathbf{n}_\Gamma)\cdot\mathbf{n}_\Gamma]_\Gamma$ from the momentum equation. It can be treated in different ways, either considering WDF or JCF. This distinction applied to the viscosity jump has been clarified in the recent work of Lalanne et al. [105].

5.3.1 Whole domain formulation

The sharp treatment of such a discontinuity is not straightforward and has been addressed using a WDF in most of the early works in VOF [104], SLS [214], ACLS [48] and CLSVOF [213].

This approach requires considering the complete stress tensor discretization with cross derivatives implying the coupling of all velocity components. For clarity, only the 2D formulation of the tensor is detailed here.
The stress tensor gives

$$\mathbf{D} = \begin{pmatrix} D^{11} & D^{12} \\ D^{21} & D^{22} \end{pmatrix} = \begin{pmatrix} 2\frac{\partial u}{\partial x} & \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} \\ \frac{\partial u}{\partial y} + \frac{\partial v}{\partial x} & 2\frac{\partial v}{\partial y} \end{pmatrix} \quad . \tag{5.26}$$

Then, a discretization of $2\nabla \cdot (\mu \mathbf{D})$ requires to determine $\frac{\partial\left(2\mu D^{11}\right)}{\partial x}$, $\frac{\partial\left(2\mu D^{12}\right)}{\partial y}$ in $\Sigma_{u,i-\frac{1}{2},j}$ and $\frac{\partial\left(2\mu D^{21}\right)}{\partial x}$, $\frac{\partial\left(2\mu D^{22}\right)}{\partial y}$ in $\Sigma_{v,i,j-\frac{1}{2}}$

$$\left.\frac{\partial\left(2\mu D^{11}\right)}{\partial x}\right|_{i-\frac{1}{2},j} = 2\frac{\mu_{i,j} D^{11}_{i,j} - \mu_{i-1,j} D^{11}_{i-1,j}}{\Delta x} \quad , \tag{5.27}$$

$$\left.\frac{\partial\left(\mu D^{12}\right)}{\partial y}\right|_{i-\frac{1}{2},j} = \frac{\mu_{i-\frac{1}{2},j+\frac{1}{2}} D^{12}_{i-\frac{1}{2},j+\frac{1}{2}} - \mu_{i-\frac{1}{2},j-\frac{1}{2}} D^{12}_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta y} \quad , \tag{5.28}$$

$$\left.\frac{\partial\left(\mu D^{21}\right)}{\partial x}\right|_{i,j-\frac{1}{2}} = \frac{\mu_{i+\frac{1}{2},j-\frac{1}{2}} D^{21}_{i+\frac{1}{2},j-\frac{1}{2}} - \mu_{i-\frac{1}{2},j-\frac{1}{2}} D^{21}_{i-\frac{1}{2},j-\frac{1}{2}}}{\Delta x} \quad , \tag{5.29}$$

$$\left.\frac{\partial\left(2\mu D^{22}\right)}{\partial y}\right|_{i,j-\frac{1}{2}} = 2\frac{\mu_{i,j} D^{22}_{i,j} - \mu_{i,j-1} D^{22}_{i,j-1}}{\Delta y} \quad . \tag{5.30}$$

values of μ are needed at $\Sigma_{\rho,i,j}$ corners, which can be defined with simple average of the neighbour collocated values [214]

$$\mu_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{1}{4}\left(\mu_{i,j} + \mu_{i-1,j} + \mu_{i,j-1} + \mu_{i-1,j-1}\right) \tag{5.31}$$

other authors proposed more complex interpolations based on the signed distance ϕ as in [215] where the μ values at the corners are computed from node fractions defined at the corners $\theta_{i-\frac{1}{2},j-\frac{1}{2}}$ such that

$$\theta_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{|\phi_{i,j}|_+ + |\phi_{i-1,j}|_+ + |\phi_{i,j-1}|_+ + |\phi_{i-1,j-1}|_+}{|\phi_{i,j}| + |\phi_{i-1,j}| + |\phi_{i,j-1}| + |\phi_{i-1,j-1}|} \quad . \tag{5.32}$$

The viscosity is then retrieved using the harmonic average

$$\mu_{i-\frac{1}{2},j-\frac{1}{2}} = \frac{\mu_l \mu_g}{\mu_g \theta_{i-\frac{1}{2},j-\frac{1}{2}} + \mu_l (1 - \theta_{i-\frac{1}{2},j-\frac{1}{2}})} \quad . \tag{5.33}$$

In [106], the authors showed that in 1D, this average leads to the GFM discretization of the gradient jump $\left[\mu \frac{\partial u}{\partial x}\right]_\Gamma = 0$. Indeed, expressions in Eqs. (5.33) and (A.15) are identical. However, this is no longer true for multi-dimensional discretizations when the interface

is not aligned with a mesh direction. Harmonic discretizations have been used in several works to deal with viscosity [81; 61; 142]
Overall, this methodology allows handling the viscosity discontinuity sharply without further treatments. The main drawback of this formulation is the difficulty to use an implicit formulation for diffusion. Indeed, as the extra-diagonal terms of $\mathbf{D}$ imply coupling between all components, the resulting implicit system is more complex to solve.

5.3.2 Jump condition formulation

Another way to handle this discontinuity is due to Kang et al. [97] where only the Laplacian of velocity is considered in the prediction step using the JCF. This term comes from the incompressible one-phase flow where $\mu\mathbf{D}$ reduces to $\mu\Delta\mathbf{u}$. However, gradients $\nabla\mathbf{u}$ are discontinuous at the interface, and discretization of the Laplacian using directly the second-order derivative $\frac{\partial^2 u}{\partial x^2}$, $\frac{\partial^2 u}{\partial y^2}$, $\frac{\partial^2 v}{\partial x^2}$ and $\frac{\partial^2 v}{\partial y^2}$ is not appropriate. One way around is to rewrite this operator as $\nabla\cdot(\mu\nabla\mathbf{u})$ and apply the resulting viscous flux jumps

$$\mathbf{J} = \begin{pmatrix} \left[\mu\frac{\partial u}{\partial x}\right]_\Gamma & \left[\mu\frac{\partial u}{\partial y}\right]_\Gamma \\ \left[\mu\frac{\partial v}{\partial x}\right]_\Gamma & \left[\mu\frac{\partial v}{\partial y}\right]_\Gamma \end{pmatrix} = \begin{pmatrix} \mathbf{n} \\ \mathbf{t} \end{pmatrix}^\intercal [\mu]_\Gamma \begin{pmatrix} \frac{\partial u_n}{\partial n} & \frac{\partial u_n}{\partial t} \\ -\frac{\partial u_n}{\partial t} & \frac{\partial u_t}{\partial t} \end{pmatrix} \begin{pmatrix} \mathbf{n} \\ \mathbf{t} \end{pmatrix} \quad , \tag{5.34}$$

with $\mathbf{t}$ the unit vector tangent to the interface. The notation $\frac{\partial u_n}{\partial t}$ stands for the $(\nabla u\cdot\mathbf{n}, \nabla v\cdot\mathbf{n})\cdot\mathbf{t}$. The idea behind this formulation is that all the derivatives in Eq. (5.34) in the orthonormal coordinate system $(\mathbf{n},\mathbf{t})$ are continuous across the interface if no phase-change occurs. A complete description of this argument is detailed in 3D in [163] with an extension to phase-change applications.
Then the GFM is applied to define the discretization of $\nabla\cdot(\mu\nabla\mathbf{u})$ close to the interface using Eq. (A.14) with $\Phi = u$, $\beta = \mu$, $a_\Gamma = 0$ and $b_\Gamma = J^{11}$.
This leads to the following expression if $\Sigma_{u,i-\frac{1}{2},j}$ is liquid and $\Sigma_{u,i+\frac{1}{2},j}$ is gas

$$\mu\nabla u|_{i,j} = \widehat{\mu}\frac{u_{i+\frac{1}{2},j} - u_{i-\frac{1}{2},j}}{\Delta x} - \frac{\theta\widehat{\mu}}{\mu_l}J^{11} \quad , \tag{5.35}$$

with $\widehat{\mu}$ the effective viscosity defined using Eq. (A.15). Finally, the jump in stress tensor is taken into account in the pressure jump through Eq. (5.3) by imposing

$$[P]_\Gamma = [(\mathbf{D}\cdot\mathbf{n}_\Gamma)\cdot\mathbf{n}_\Gamma]_\Gamma = 2\,[\mu]_\Gamma\,\frac{\partial u_n}{\partial n} \quad . \tag{5.36}$$

This is done using the GFM approach again.
The implicit treatment is easier using this approach as the velocity component are decoupled and solved by independent linear systems.

5.3.3 Hybrid formulation

The last approach is due to Lalanne et al. [105] by combining the relations

$$\nabla H_\Gamma = \delta_\Gamma\mathbf{n}_\Gamma \quad , \tag{5.37}$$

and

$$\mu = \mu_g + H_\Gamma^\epsilon \left[\mu\right]_\Gamma \quad , \tag{5.38}$$

to obtain

$$\nabla\mu = \nabla\left(\mu_g + \left[\mu\right]_\Gamma H_\Gamma\right) = \left[\mu\right]_\Gamma \nabla H_\Gamma = \left[\mu\right]_\Gamma \delta_\Gamma \mathbf{n}_\Gamma \quad . \tag{5.39}$$

Then, the viscous operator can be written as

$$\begin{aligned}\nabla\cdot\left(\mu\mathbf{D}\right) &= \nabla\cdot\left(\mu\left(\nabla\mathbf{u} + \nabla^\intercal\mathbf{u}\right)\right)\\ &= \nabla\cdot\left(\mu\nabla\mathbf{u}\right) + \mu\nabla\cdot\nabla^\intercal\mathbf{u} + \nabla\mu\cdot\nabla^\intercal\mathbf{u}\\ &= \nabla\cdot\left(\mu\nabla\mathbf{u}\right) + \left[\mu\right]_\Gamma\frac{\partial u_n}{\partial n} \quad .\end{aligned} \tag{5.40}$$

This last result is used to create a new splitting, the viscous operator is taken as $\nabla\cdot\left(\mu\nabla\mathbf{u}\right)$ and the pressure jump $\left[\mu\right]_\Gamma\frac{\partial u_n}{\partial n}$ is imposed through Eq. (5.3).
The main advantage of such an approach is the straightforward implicit implementation and application to flows with phase-change. However, a timestep limitation is still due to viscosity because of the jump imposed in the Poisson equation. This new constraint still scales in Δx, but it depends on the jump in viscosity which is less restrictive.
This method is referred to as Hybrid Whole Domain Jump Condition Formulation (HWDJCF) in the following.

5.3.4 Continuous equivalence

In [105], a framework is described where equivalence between the three approaches described above is proven. Let us write the splitting imposed by a projection method, only considering the viscous contribution for clarity (it still holds if convection or gravity is included), the following expression of the prediction can be expressed

$$\mathbf{u}_1^* = \mathbf{u}^n + \frac{\nabla\cdot\left(\mu\mathbf{D}\right)}{\rho} \quad , \tag{5.41}$$

$$\mathbf{u}_2^* = \mathbf{u}^n + \frac{\mu\Delta\mathbf{u}}{\rho} \quad , \tag{5.42}$$

$$\mathbf{u}_3^* = \mathbf{u}^n + \frac{\nabla\cdot\left(\mu\nabla\mathbf{u}\right)}{\rho} \quad , \tag{5.43}$$

with $\mathbf{u}_1^*$ the WDF predicted velocity, $\mathbf{u}_2^*$ the JCF predicted velocity and $\mathbf{u}_3^*$ the HWDJCF predicted velocity. Then the correction step is written as

$$\mathbf{u}_1^{n+1} = \mathbf{u}_1^* - \frac{\Delta t}{\rho}\nabla P^{n+1} \quad , \tag{5.44}$$

$$\mathbf{u}_2^{n+1} = \mathbf{u}_2^* - \frac{\Delta t}{\rho}\left(\nabla P^{n+1} - 2\left[\mu\right]_\Gamma\frac{\partial u_n}{\partial n}\delta_\Gamma\mathbf{n}_\Gamma\right) \quad , \tag{5.45}$$

$$\mathbf{u}_3^{n+1} = \mathbf{u}_3^* - \frac{\Delta t}{\rho}\left(\nabla P^{n+1} - \left[\mu\right]_\Gamma\frac{\partial u_n}{\partial n}\delta_\Gamma\mathbf{n}_\Gamma\right) \quad . \tag{5.46}$$

By noticing that $\nabla \cdot (\mu \mathbf{D}) = \mu \Delta \mathbf{u} + 2 \left[\mu\right]_\Gamma \frac{\partial u_n}{\partial n} \delta_\Gamma \mathbf{n}_\Gamma = \nabla \cdot (\mu \nabla \mathbf{u}) + \left[\mu\right]_\Gamma \frac{\partial u_n}{\partial n}$, then $\mathbf{u}_1^{n+1} = \mathbf{u}_2^{n+1} = \mathbf{u}_3^{n+1}$.
The three above approaches are then different choices of splitting the viscosity jump and the viscous operator in the projection method. The complete comparison of these approaches is provided in [105]. WDF and JCF were found to be more accurate than the hybrid formulation.
In light of these conclusions, a WDF is used in the solver as it is easy to implement and leads to accurate results. Moreover, the loss of the viscous timestep constraints using an implicit formulation does not necessarily lead to a gain in efficiency as a restrictive constraint related to surface tension prevails for coarse to medium meshes.

5.4 Surface tension effects

The pressure jump related to surface tension can be treated as a volumetric force $\mathbf{f}_\sigma$ in the momentum equation. This new term is expressed as

$$\mathbf{f}_\sigma = \sigma \kappa \delta_\Gamma \mathbf{n}_\Gamma \quad . \tag{5.47}$$

Note that only the volumetric formulation is presented here, but another class of methods uses the integral form of the surface tension [222] and includes directly Marangoni stresses. In [2], a momentum conservative well-balanced surface tension model is introduced in the Level-Set framework using this method.
A surface tension modelling requires both an accurate curvature computation and a consistent definition of $\delta_\Gamma \mathbf{n}_\Gamma$. These two numerical aspects are detailed hereafter.

5.4.1 General formulation of the surface tension

The literature often presents Continuum Surface Force (CSF) and Ghost Fluid Method (GFM) as two different paradigms:

- CSF is relying on a WDF;
- GFM is based on a JCF;

The CSF was first introduced by Brackbill et al. [22] in a diffuse VOF framework where the surface tension contribution is defined as

$$\mathbf{f}_\sigma = \sigma \kappa \nabla \widetilde{f} \quad , \tag{5.48}$$

with $\widetilde{f}$ a smoothed version of the volume fraction f. In their paper, they noticed that smoothing of f was improving the results. This was partly due to the fact that curvature was also deduced from differentiation of $\widetilde{f}$. As pointed out in Section 5.4.2.1, this is a strategy that improves results for curvature. However, there are no clear evidence that using $\widetilde{f}$ instead of f in the gradient operator improves the results. This original paper does not provide more information about this point as they use a global operator for $\mathbf{F}_\sigma$ including both computation of κ and $\delta_\Gamma \mathbf{n}_\Gamma$.
The GFM presented in Appendix A.1 can be applied to the pressure equation. By identification of Eqs. (5.3) and (A.2), $\Phi = P$, $\beta = \frac{1}{\rho}$, $a = [P]_\Gamma$ and $b = 0$.

Then in the 1D example, a second-order pressure gradient computation at $i-\frac{1}{2}$ with gas at the left and liquid at the right leads to

$$\left.\frac{\partial P}{\partial x}\right|_{i-\frac{1}{2}} = \frac{P_i - P_{i-1}}{\Delta x} - \frac{[P]_\Gamma}{\Delta x} \quad . \tag{5.49}$$

In the reverse case, with liquid at the left and gas at the right of the interface, the same derivation leads to

$$\left.\frac{\partial P}{\partial x}\right|_{i-\frac{1}{2}} = \frac{P_i - P_{i-1}}{\Delta x} + \frac{[P]_\Gamma}{\Delta x} \quad . \tag{5.50}$$

This method allows defining a sharp representation of the pressure jump and a direct way to take into account surface tension.
Some authors tried to sharpen the CSF method in a VOF framework like in the Sharp Surface Force (SSF) proposed in [62]. On the other hand, the first LS methods used a Delta function approach relying on smoothed Heaviside from the regularization of a distance function like in [56]. However, by rewriting the two above formulations, Popinet [170] showed that the GFM method can be rewritten into a volumetric force formulation just as CSF

$$\boxed{\mathbf{f}_\sigma = \sigma\kappa\nabla H_\Gamma} \quad . \tag{5.51}$$

5.4.1.1 Heaviside definitions

The only difference between the original CSF method and the GFM method is the choice of $\delta_\Gamma \mathbf{n}_\Gamma$ computation in Eq. (5.47). By using Eq. (5.37), it results in a choice of Heaviside H_Γ as already presented in Chapter 2.
For the original CSF method, identification in Eq. (5.48) gives $H_\Gamma = \tilde{f}$. For the GFM, the equivalence in Eq. (A.22) leads to the sharp Heaviside related to JCF

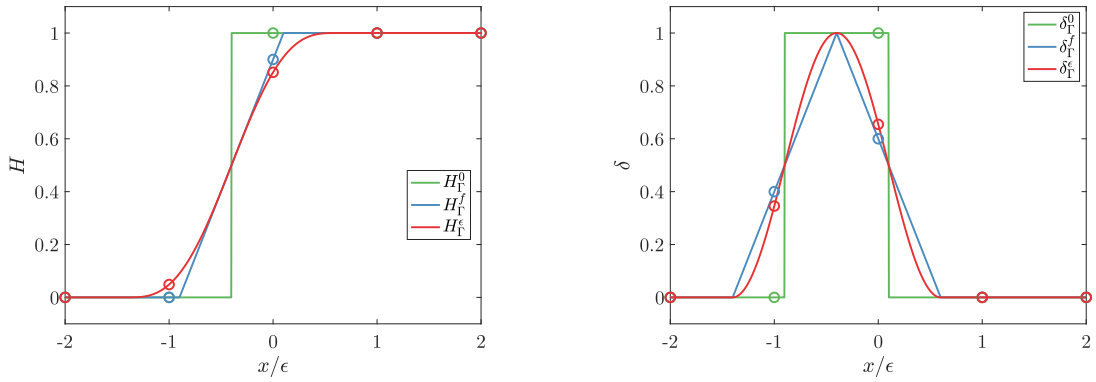
$$H_\Gamma^0 = \begin{cases} 0 & \text{if } \mathbf{x} \in \Omega_g \\ 1 & \text{if } \mathbf{x} \in \Omega_l \end{cases} \quad . \tag{5.52}$$

Note that the SSF method uses exactly the same definition in a VOF framework.
In a VOF framework, the natural Heaviside is the volume fraction $H_\Gamma^\epsilon = f$ while in a Level-Set framework, a Heaviside can be defined using the smooth Heaviside H_Γ^ϵ defined by Eq. (2.37). In the conservative Level-Set context, the natural choice is $H_\Gamma^\epsilon = \psi$. They are then associated with a WDF.

5.4.1.2 Resolution

An interesting observation of Popinet in his review [170] is that there is no resolution difference between the smooth and sharp Heaviside choices if the smoothing width is defined such that the interface width $\epsilon \leq \Delta x$.
Indeed, the Dirac definition in the case of a GFM formulation is insensitive to a shift of

$\frac{\epsilon}{2}$ to the left or the right. Thus, even if it is sharper in its representation, it produces a method with the exact same resolution as the CSF formulation. An illustration of this is shown in Fig. 5.6



(a) *Heaviside function* **(b)** *Dirac distribution*

Figure 5.6: *Jump resolution for different Heaviside H_Γ choices. The lines represent the continuous values while the points are the mesh values.*

5.4.1.3 Well-balanced property

One important aspect of surface tension modelling is its capacity to maintain an equilibrium for the Laplace problem. Many numerical methods proposed in the literature suffer from parasitic currents when simulating a static droplet at equilibrium. In the worst case, spurious currents are amplified when iterating in time. This is the case for the original CSF method [22].
In order to understand why this does happen, let us write the momentum equation for the static droplet (zero velocity, constant curvature κ and surface tension σ)

$$-\nabla P + \kappa\sigma\nabla H_\Gamma = 0 \quad . \tag{5.53}$$

Renardy & Renardy noted in [178] that an exact balance can be achieved if one uses the same face gradient operator ∇ for pressure and Heaviside. The discrete Laplace equilibrium in a MAC grid arrangement leads to

$$\nabla\left(\kappa\sigma H_\Gamma - P\right) = 0 \quad . \tag{5.54}$$

Integration of this last equation gives the classical Laplace pressure

$$P = \sigma\kappa H_\Gamma + C \quad , \tag{5.55}$$

with C a constant. Such a strategy was not used in most early works on surface tension modelling, especially in the CSF approaches.
In [97], authors noticed that GFM reduces the parasitic currents significantly in the case of

the Laplace equilibrium. They argued that it comes from the sharper representation of the pressure jump compared to a CSF approach. In fact, the smearing is not the direct cause of parasitic currents. The difference lies in the well-balanced properties of the schemes. While GFM is naturally well-balanced in its original formulation, CSF approaches was not.
Note that if κ is not constant discretely because of numerical errors, then Eq. (5.53) cannot be simplified to Eq. (5.54) and exact balance is not expected anymore. This is discussed further in Section 6.2.1.2.
Thus a consistent discretization of pressure and Heaviside leads to robust and well-balanced surface tension modelling only with an accurate curvature computation.

5.4.2 Curvature computation

As mentioned before, the main aspect of surface tension modelling is the computation of the curvature. It is done by using the interface information provided by the color function. The two main approaches of the literature use either direct differentiation of the implicit function representing the interface with standard operators or the explicit interface topology.

5.4.2.1 Curvature from implicit surfaces

A natural choice for curvature computation is the differentiation of the field c. In the context of implicit surfaces, Goldman [70] provided an exhaustive list of formulas for curvature computation. From c, the normal pointing outward the liquid phase can be computed as

$$\mathbf{n}_\Gamma = -\frac{\nabla c}{|\nabla c|} \quad . \tag{5.56}$$

The curvature is then deduced from one of the following expressions

$$\kappa = -\nabla \cdot \left(\frac{\nabla c}{|\nabla c|} \right) \quad , \tag{5.57}$$

$$\kappa = \frac{\mathrm{tr}\left(\nabla\nabla c\right) - \mathbf{n}_\Gamma \cdot \nabla\nabla c \cdot \mathbf{n}_\Gamma^\intercal}{|\nabla c|} \quad . \tag{5.58}$$

Order requirement As curvature depends on second-order derivatives of the function c, one needs to provide c at least at third order to retrieve a consistent curvature using standard finite difference operators. More generally, a curvature of order m is obtained using an implicit surface function of order $m+2$ [41]. This limitation is important to keep in mind, as numerous strategies to reconstruct a distance function from an isocontour are of second-order, which is not sufficient to ensure convergence of the curvature. This has been observed in a VOF context [42] with curvature computed from a second-order Reconstructed Distance Function (RDF). To deal with this issue, a least-square computation of operators has been presented in the context of curvature computation from an unstructured mesh [125] or a FMM reconstruction of a signed distance [38]. The idea is to reduce the second-order errors of the distance function by adding more points in the stencil. By using a Taylor expansion around a point of interest $\mathbf{x}_\mathcal{C}$

$$c(\mathbf{x}) = c(\mathbf{x}_\mathcal{C}) + \Delta\mathbf{x}^\intercal \nabla c(\mathbf{x}_\mathcal{C}) + \Delta\mathbf{x}^\intercal \nabla\nabla c(\mathbf{x}_\mathcal{C}) \Delta\mathbf{x} \quad , \tag{5.59}$$

with $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_\mathcal{C}$. This leads to a second-order gradient operator ∇c and a first-order Hessian operator $\nabla\nabla c$ at $\mathbf{x}_\mathcal{C}$. A box stencil of 3×3 or 5×5 is used to construct the following least square linear system

$$\begin{pmatrix} 1 & \Delta x_1 & \Delta y_1 & \frac{1}{2}\Delta x_1^2 & \frac{1}{2}\Delta y_1^2 & \Delta x_1\Delta y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta x_N & \Delta y_N & \frac{1}{2}\Delta x_N^2 & \frac{1}{2}\Delta y_N^2 & \Delta x_N\Delta y_N \end{pmatrix} \begin{pmatrix} c(\mathbf{x}_\mathcal{C}) \\ \frac{\partial c}{\partial x}(\mathbf{x}_\mathcal{C}) \\ \frac{\partial c}{\partial y}(\mathbf{x}_\mathcal{C}) \\ \frac{\partial^2 c}{\partial x^2}(\mathbf{x}_\mathcal{C}) \\ \frac{\partial^2 c}{\partial y^2}(\mathbf{x}_\mathcal{C}) \\ \frac{\partial^2 c}{\partial x\partial y}(\mathbf{x}_\mathcal{C}) \end{pmatrix} = \begin{pmatrix} c(\mathbf{x}_1) \\ \vdots \\ c(\mathbf{x}_N) \end{pmatrix} \quad . \tag{5.60}$$

This linear system requires at least $N = 6$ (10 in 3D) to be well-posed. Note that a weight matrix can also be introduced in order to restrain the least square to the point of interest

$$W_{ii} = \exp\left(-\left(\frac{2\|\Delta\mathbf{x}\|}{3\Delta_m}\right)^2\right) \quad , \tag{5.61}$$

with Δ_m the mesh size. Even if the method leads to second-order convergence in a coarse regime, however, it will be demonstrated from numerical results that the convergence is not retrieved for high resolutions in Sec. 6.2.1.1.

Regularity requirement Another requirement is the smoothness of c to avoid spurious higher-order derivatives. With LS, this is not a problem, as ϕ is already a smooth, well-defined function in the entire domain. However, for VOF, the volume fraction is too sharp to provide non-oscillating curvature computation. Using directly the volume fraction f leads to non-converging curvature. To improve the behaviour, various convolution methods were used to retrieve a smoother version $\tilde{f}$ (see [230]). For a given point $\mathbf{x}_\mathcal{C}$, the general convolution procedure can be written as

$$\tilde{f}(\mathbf{x}_\mathcal{C}) = K * f(\mathbf{x}_\mathcal{C}) = \int K\left(\mathbf{x} - \mathbf{x}_\mathcal{C}\right) f(\mathbf{x}) d\mathbf{x} \quad , \tag{5.62}$$

with K a convolution Kernel that can be defined in various ways.
It has been shown that even with 8^{th} order convolution Kernel, curvature cannot reach a spatial convergence. However, results are improved with the method.

Large stencil Finally, it is not always interesting to look for higher stencils with standard finite difference operators in the case of fast varying c. It happens when two interfaces are really close to each other, or when an interface topology is under-resolved. In such cases, large stencils will amplify high frequencies errors and provide a poorer curvature approximation. This issue is addressed in [57] where close interfaces are separated, and local reconstruction of a signed distance belonging to only one of the front is performed before computing the normal and curvature.

Considering the above discussion, differentiation seems to be a straightforward method for SLS, ACLS or CLSVOF. Simple second-order finite differences are sufficient in the SLS case, while a least-square approach is mandatory for ACLS and CLSVOF to have convergence. However, a more reliable method is needed for curvature from VOF, as f cannot be reasonably differentiated twice.

5.4.2.2 Curvature from height function

Another alternative to computing curvature is the construction of height function (HF) from the indicator function. This has been widely used in the VOF context as standard operator fail to give a converging curvature. Height functions are based on the idea that if an interface is described by a function $\hbar$ such that $y = \hbar(x)$ in 2D (or $z = \hbar(x, y)$ in 3D), the curvature can be deduced from

$$\kappa = \frac{\hbar_{xx}}{(1+\hbar_x^2)^{\frac{3}{2}}} \qquad \text{in 2D} \quad , \tag{5.63}$$

$$\kappa = \frac{\hbar_{xx} + \hbar_{yy} + h_{xx}\hbar_y^2 + \hbar_{yy}\hbar_x^2 - 2\hbar_{xy}\hbar_x\hbar_y}{(1+\hbar_x^2+\hbar_y^2)^{\frac{3}{2}}} \qquad \text{in 3D} \quad . \tag{5.64}$$

If the height function is defined exactly, then κ can be computed at any order depending on the gradient and Hessian operators. This method has the advantage of naturally computing the curvature at the interface location in opposition to the standard operator method. However, it is important to remember that curvature at second-order accuracy can only be achieved with an exact (or at least fourth order) volume fraction field. It is never the case when dealing with dynamic interfaces, as the VOF geometric schemes are often limited to second-order accuracy (see test case of Section 6.2.1.4).
Note that height functions can also be used to compute normal (see [60]), a complete presentation of the height function method and its application (with higher-order computation) is presented in [19].

Fixed stencil An early method was proposed by Sussman et al. [210] in the context of a height function computed from volume fraction. In a cell $\mathcal{C}$, the dominant direction is determined from the normal $\mathbf{n}$. Let us take a 2D case where $n_y > n_x$ then an exact average height function h_i can be derived by simply summing all volume fraction of a column in the y direction

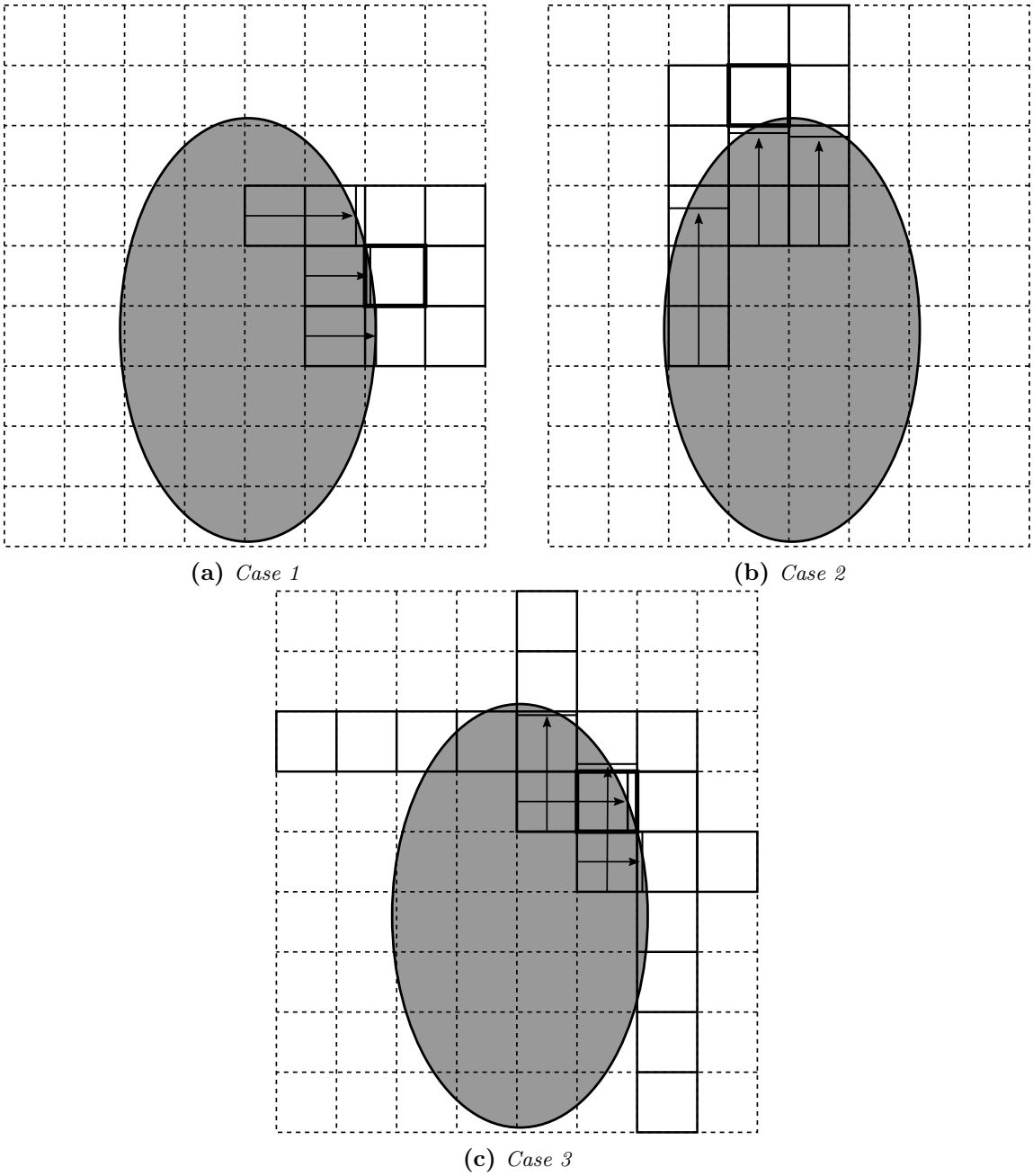
$$h_i = \sum_{j=-\infty}^{j=+\infty} f_{i,j} \quad . \tag{5.65}$$

In this method ∞ is obviously not possible and replaced by 3.
The operators h_x and h_{xx} are then computed with second-order accuracy using standard finite differences $h_x|_i = \frac{h_{i+1}-h_{i-1}}{2\Delta x}$ and $h_{xx}|_i = \frac{h_{i+1}-2h_i+h_{i-1}}{\Delta x^2}$.
Finally, the curvature is retrieved with second-order accuracy using Eq. (5.64). This method requires a 7 points stencil in the height function direction and 3 points in the other directions leading to a 3×7 ($3 \times 3 \times 7$ in 3D) stencil.

Adaptive stencil The fixed stencil will not work in any interface configurations, this issue is even more visible in under-resolved configurations.
In the first case represented in Fig. 5.7a the 3×7 would have been enough to retrieve a second-order curvature approximation. Indeed all three height functions are consistent. This means that the base of the cell at the bottom of the height is full ($f = 1$), and the cell at the top is empty ($f = 0$).
In the second case, in Fig. 5.7b, a 9-stencil is needed to retrieve a consistent height function. Another problem in this configuration is that the middle height function is lower than the coordinate of the lower face of the cell in which the height function is computed. In this type of configuration, Cumins et al. reported that a good approximation of the curvature is not possible [42].
Finally, even in fairly resolved interfaces, it is not possible to have three consistent height functions in a given direction as shown in Fig. 5.7c.
The use of an adaptive stencil can solve most of the above issues. This approach has been presented in [79] where the stencil is uniform for the 3 (9 in 3D) height functions.
Another approach presented by Popinet in his solver [169] constructs each height independently with a variable stencil.

Mesh decoupling An attempt to decouple height function from the mesh was first proposed by Liovic et al. [114]. In order to deal with symptomatic configurations (such as Fig. 5.7c), they introduced a reconstruction in the diagonal direction of an orthogonal mesh in addition to the mesh directions using cell center as shown in Fig. 5.8a. It has been shown in [119] that the maximum error in the curvature computation of a sphere occurs at 45°. The method reduces the maximum error, which then occurs at an angle of 22.5°.
Later, Owkes and Desjardins [157] generalized this idea by computing height functions in a coordinate system orthonormal to the interface normal with a column of parametrized width and depth. In Fig. 5.8b, a height construction is shown with a width lower than Δx. This method is robust and does not fail in symptomatic configurations when the width is higher than $2\Delta x$.
Both approaches use geometric reconstructions based on simplex to retrieve the volume fraction contained in each height function.

Level-Set Height function Even if height functions are mainly used and well-designed for VOF representation of the interface, some attempts to apply it has been made in Level-Set (see [212] or [119]). There are advantages of using height function approach instead of the standard operator approach: the curvature is computed at the interface location, the computation is more robust in the context of two close interfaces as it will adapt the stencil to take it into account.
The algorithm is very similar to the one used for a volume fraction field. The height functions are build by going through a given stencil below and above the cell of interest. The stencil is adapted and stops when a new sign change occurs or the maximum stencil is reached. It is shown that a maximum stencil of 3 is enough to have converging curvature.
In the context of Conservative Level-Set, Owkes et al. [155] proposed to apply height function method to the function ψ. They manage to obtain curvature convergence by using

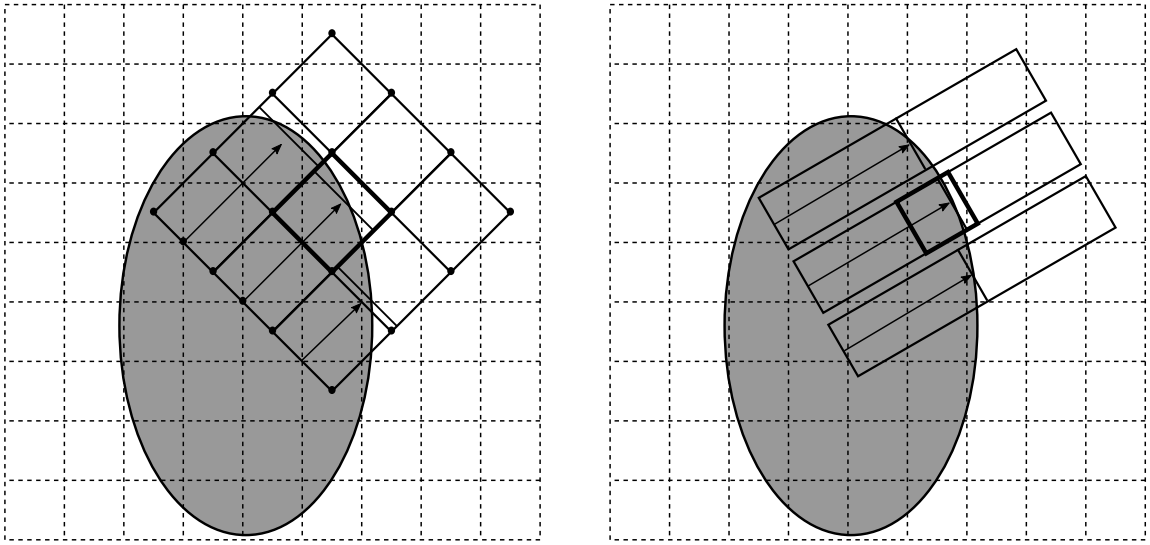


(a) *Case 1* **(b)** *Case 2*

(c) *Case 3*

Figure 5.7: *Different configurations of HF computation. The arrows represent the hight function while the plane lines represent the adaptive stencil used to reconstruct them. The case of Fig. 5.7c presents non-consistent HF reconstructions.*

very high stencil of 3. The explanation of this stencil requirement is that the function ψ has an interface thickness. If the interface thickness is $\epsilon = 0.5\Delta x$ then the interface is captured over 4 cells. They then propose a formula to determine a sufficient stencil $S = 7 + 8\frac{\epsilon}{\Delta x}$.



(a) *Liovic et al.* **(b)** *Owkes et al.*

Figure 5.8: *Mesh decoupled HF. The arrows represent the hight function while the plane lines represent the adaptive stencil used to reconstruct them.*

The use of standard height function with a variable stencil instead of the mesh-decoupled method proposed in [157] benefits from the Cartesian grid arrangement. It is then a promising algorithm for curvature computation in a VOF framework. However, another computation method is required for under-resolved configurations following the idea in [169] detailed in the next section.

5.4.2.3 Curvature from interface positions

A last type of method, mainly used in Front Tracking methods, relies on marker positions to reconstruct an interface surface using a least square approach [69].
For the general case of an interface with normal $\mathbf{n}$ and a set of points describing the interface $\mathbf{x}_{\Gamma,i}$, one can fit a parabola $\mathcal{P}$ by using a least square regression.

$$\mathcal{P}(\mathbf{x}) = a_0 x^2 + a_1 x + a_2 \qquad \text{in 2D} \quad , \tag{5.66}$$

$$\mathcal{P}(\mathbf{x}) = a_0 x^2 + a_1 y^2 + a_2 xy + a_3 x + a_4 y + a_5 \qquad \text{in 3D} \quad . \tag{5.67}$$

First, the set of points $\mathbf{x}_{\Gamma,i}$ is redefined as $\mathbf{x}'_{\Gamma,i}$ in a new coordinate system $[\mathbf{n}, \mathbf{t}_1, \mathbf{t}_2]$ from the Cartesian coordinate system $[\mathbf{e}_x, \mathbf{e}_y, \mathbf{e}_z]$.
Then, the following least square minimization is performed

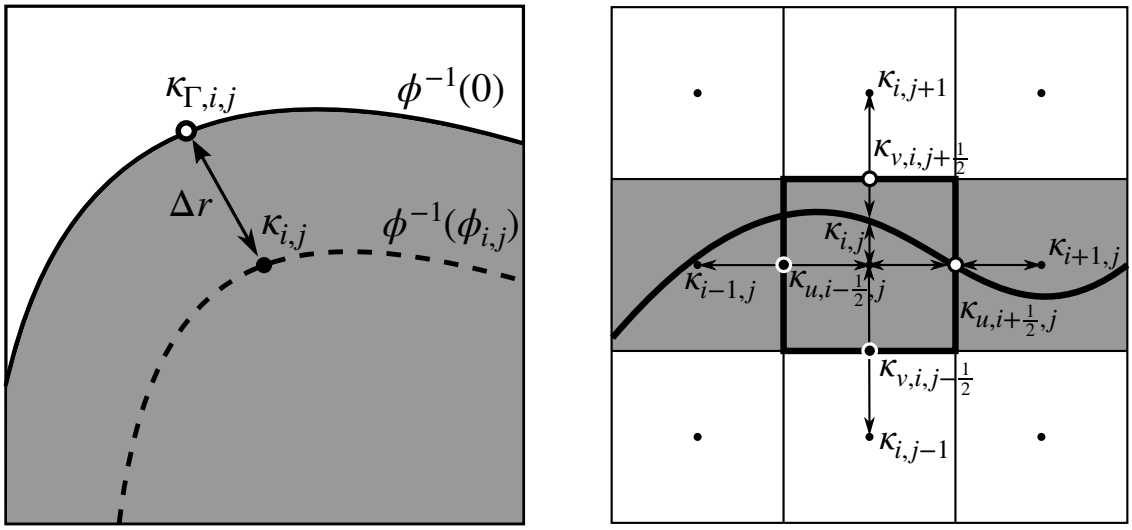
$$R^2 = \sum_{i=1}^{N} W_i(\mathbf{x}'_{\Gamma,i}) \left(z'_{\Gamma,i} - \mathcal{P}(\mathbf{x}'_{\Gamma,i}) \right)^2 \quad , \tag{5.68}$$

with W_i an optional weight to restrain the region of interest.
This resulting linear system requires at least $N = 3$ (6 in 3D) to be well-posed.

Even if this method seems expensive and complex to implement,this is a good alternative to the mesh-decoupled method of [157] for robust under-resolved computations. It has been used in a VOF context in [154] with positions from the PLIC reconstruction leading to strong improvement in under-resolved configurations compared to methods based on height function. As proposed in [169], this parabola fit can be coupled with height function to improve under-resolved robustness while keeping a second-order convergence in high resolution for a VOF framework.

5.4.2.4 Curvature interpolation to the face

Depending on the method, the curvature is computed either at the interface or at the cell center as illustrated in Fig. 5.9a. $\kappa_{\Gamma,i,j}$ is directly the curvature of the interface lying in $\mathcal{C}_{i,j}$ while $\kappa_{i,j}$ is the curvature of an interface defined by the isocontour $c^{-1}(c_{i,j})$ which is not $f^{-1}(0.5)$, $\phi^{-1}(0)$ or $\psi^{-1}(0.5)$.
In a MAC grid arrangement, these curvature values need to be interpolated to the face as κ_f. The way to do it depends on the location of the computed curvature.



(a) *Curvature position* **(b)** *Curvature interpolation*

Figure 5.9: *Position and interpolation of curvature computed from ϕ in a cell $\mathcal{C}_{i,j}$ with $\kappa_{i,j}$ the curvature defined at the cell center and $\kappa_{\Gamma,i,j}$ the curvature of the interface contained in the cell. The grey cells contain the interface while white cells are either liquid or gas cells. White points ($\kappa_{i+\frac{1}{2},j}$ and $\kappa_{i,j+\frac{1}{2}}$) are the face curvature where an interface crosses the stencil of interpolation. Black points ($\kappa_{i-\frac{1}{2},j}$ and $\kappa_{i,j-\frac{1}{2}}$) are face interpolation from two curvature points of the same phase.*

Curvature defined at the cell center When curvature is defined at the center of a cell, a value is contained in all domain cells (white and grey cells of Fig. 5.9b). The curvature

interpolation can be expressed as

$$\kappa_{u,i-\frac{1}{2},j} = \frac{\kappa_{i,j}|\phi_{i-1,j}| + \kappa_{i-1,j}|\phi_{i,j}|}{|\phi_{i,j}| + |\phi_{i-1,j}|} \qquad \text{for linear interpolation,} \tag{5.69}$$

$$\kappa_{u,i-\frac{1}{2},j} = \kappa_{i,j}\kappa_{i-1,j}\frac{|\phi_{i,j}| + |\phi_{i-1,j}|}{\kappa_{i,j}|\phi_{i,j}| + \kappa_{i-1,j}|\phi_{i-1,j}|} \qquad \text{for harmonic interpolation.} \tag{5.70}$$

The first one is the linear interpolation of the curvature, while the second one is the linear interpolation of the radius. This type of interpolation can be seen as the interpolation of cell-centered curvature to the interface position lying between these cells which can be different from the position of the face center (this is the case between $\mathbf{x}_{i,j}$ and $\mathbf{x}_{i,j+1}$). If the Heaviside is smoother than the GFM one, $\nabla H|_{i-\frac{1}{2},j} \neq 0$, hence a value κ_f is also needed, and can be computed as a simple average

$$\kappa_{u,i-\frac{1}{2},j} = \frac{\kappa_{i-1,j} + \kappa_{i,j}}{2} \quad . \tag{5.71}$$

Note that this average leads to erroneous evaluations of curvature, inducing large errors. This point will be detailed in Section 6.2.1.3.

Curvature defined at the interface For a curvature directly defined at the interface, one needs to average the curvatures of 2 adjacent cells which are not necessarily containing a curvature value .
The most simple way to do it is an average of values if they both contain an interface value (grey cells of Fig. 5.9b) and keep the single available value if one of the two adjacent cells does not contain the interface (white cells of Fig. 5.9b).

$$\kappa_{u,i-\frac{1}{2},j} = \frac{\kappa_{\Gamma,i-1,j} + \kappa_{\Gamma,i,j}}{2} \qquad \text{for two adjacent interfacial cells} \quad , \tag{5.72}$$

$$\kappa_{u,i-\frac{1}{2},j} = \kappa_{\Gamma,i,j} \qquad \text{for one interfacial cell} \quad . \tag{5.73}$$

Note that if both cells are empty, curvature cannot be computed with this approach, and one needs to obtain it from normal extrapolation or cell to cell propagation. However, this case never happens if one chooses a Heaviside $H_\Gamma = f$ or sharper.
In a more general way, one can define the curvature at a cell face by weight interpolation [178]

$$\kappa_{u,i-\frac{1}{2},j} = \frac{W_{i-1,j}\kappa_{\Gamma,i-1,j} + W_{i,j}\kappa_{\Gamma,i,j}}{W_{i-1,j} + W_{i,j}} \quad , \tag{5.74}$$

with $W_{i,j} = H^\epsilon_{\Gamma,i,j}(1 - H^\epsilon_{\Gamma,i,j})$. This gives importance to cell containing a large portion of interface (H^ϵ_Γ close to 0.5) and cancels quasi empty or full cells (H^ϵ_Γ close to 0 or 1) which are more prompt to curvature computation errors.

5.4.3 Key points of the surface tension modelling

The above state of the art shows the subtleties of treating surface tension in a two-phase flow solver. Some key points arise from the above considerations:

1. By using the same gradient operator for ∇P and ∇H_Γ provides a natural well-balanced method that limits spurious behaviours of the surface tension modelling.
2. Using ϕ allows accurate and straightforward evaluation of the curvature. However, it is crucial to keep in mind that ϕ from ACLS or CLSVOF suffers from second-order perturbations leading to significant curvature errors.
3. VOF needs to use more consuming methods based on geometry as HF or parabola fit to retrieve acceptable curvature evaluations for all resolutions.
4. Direct differentiation of c provides a curvature defined at the cell center and needs a particular interpolation step to be well-defined at the interface. This is not the case for HF or parabola fit evaluations of the curvature, which are naturally defined at the interface.

To conclude, a robust and accurate surface tension modelling can be derived for any interface capturing method by defining a proper gradient operator, an adapted curvature computation method and a correct interpolation procedure. It is important to remind that the choice of this modelling is highly dependant on the choice of the interface capturing method and will lead to different performances.

5.5 Conclusion

In this chapter, an overview of the different challenges inherent to the resolution of incompressible two-phase flows have been presented. Emphasis has been put on:

- Momentum transport, which has to be consistent with the mass transport in order to provide a robust solver for high-density ratios.
- Viscosity jump, which is not straightforward to treat implicitly.
- Surface tension modelling, which requires special attention for the description of the Dirac operator and for the curvature computation.

In light of the state-of-the-art, the choices of operator discretization employed in this work are summed up in Table 5.1.

The next chapter presents a unified framework where all challenging aspects of the two-phase flow simulations are treated carefully. The framework relies on the methods presented above, adapted to the four interface capturing methods in the scope of this work. Then these four methods are compared on multiple test cases of the literature.

Operator	$\mathcal{L}_{\text{pres}}$	$\mathcal{L}_{\text{conv}}$	$\mathcal{L}_{\text{visc}}$	$\mathcal{L}_{\text{cap}}$
Method	PFM	WENO5-cons (Sec. 5.2.5.4)	WDF	GFM

Table 5.1: *Summary of computation choices for the operators used to solve the momentum equation*

Chapter 6

Simulation of two-phase flows with a unified framework

Contents

The previous section presented a large variety of methods to treat interface discontinuities in low Mach solvers. However, they are usually adapted to one particular interface capturing method, and comparison to existing strategies are not systematically provided.
Detailed comparisons have been performed in the last decades, but they are still limited to a particular interface capturing method [63; 149; 203] or to a specific test case such as spurious currents [1] or rising bubbles [16]. Moreover, as the field of interface capturing methods is highly active, updating comparisons studies is required to take advantage of the recent advances. Finally, conclusions are sometimes only provided to 2D simulations [65; 1; 16]. This is a severe limitation when dealing with geometrical properties as the extension to 3D is not always straightforward. Our work is thus motivated by this observation and aims to provide a complete study of up-to-date methods in both 2D and 3D configurations. To avoid potential misleading conclusions due to solver differences, they are adapted to the same unified framework with the following specifications: a sharp transport of the interface, a well-balanced description of the surface tension and a consistent mass and momentum

transport.
In Section 6.1, the unified framework implementation in a finite-volume discretization on cartesian grids is detailed. Then, numerical results are provided to evaluate the surface tension modelling and the momentum conservation in Section 6.2. Finally, more realistic test cases are presented in Section 6.3 to asses the methodology proposed in this work.

6.1 Presentation of the unified framework

The present two-phase low-Mach solver relies on the projection method [39] with a WDF of the momentum equation. The properties are averaged such that:

$$\rho = \rho_g + H_\Gamma^\epsilon \left[\rho\right]_\Gamma \quad , \tag{6.1}$$
$$\mu = \mu_g + H_\Gamma^\epsilon \left[\mu\right]_\Gamma \quad , \tag{6.2}$$

with H_Γ^ϵ a regularized Heaviside of controlled width $2\epsilon = \Delta x$. In a VOF and ACLS framework, the natural choice is the color function (f and ψ respectively) while in a SLS framework, an Heaviside can be defined using a regularization of the distance function such as

$$H_\Gamma^\epsilon = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ 1 & \text{if } \phi > \epsilon \\ 1 + \frac{\phi}{2\epsilon} + \frac{1}{2\pi}\sin(\frac{\pi\phi}{\epsilon}) & \text{otherwise} \end{cases} \quad . \tag{6.3}$$

Note that this stands for the collocated values of ρ and μ, while staggered values or values at the corners can be needed in the following operator definitions. In that case, the discretization can differ from the collocated definition given here, and further details on the computation are provided.

6.1.1 Prediction step

The prediction step is written as follows

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = \mathcal{L}_{\text{conv}} + \mathcal{L}_{\text{visc}} + \mathcal{L}_{\text{cap}} \quad . \tag{6.4}$$

The three operators are defined through the following splitting

$$\mathcal{L}_{\text{conv}} = -\mathbf{u}^n \cdot \nabla_f \mathbf{u}^n \quad , \tag{6.5}$$
$$\mathcal{L}_{\text{visc}} = \frac{1}{\rho_f^{n+1}} \nabla_f \cdot \left(\mu^{n+1} \mathbf{D}^n\right) \quad , \tag{6.6}$$
$$\mathcal{L}_{\text{cap}} = \frac{1}{\rho_f^{n+1}} \mathbf{F}_\sigma^{n+1} \quad , \tag{6.7}$$

where ρ_f^{n+1}, μ^{n+1} and $\mathbf{F}_\sigma^{n+1}$ are computed using the advanced color function c^{n+1}. Details on the operators discretization are given hereafter.

6.1.1.1 Convection term $\mathcal{L}_{\text{conv}}$

The convection operator is treated using the consistent transport of mass and momentum presented in Section 5.2.3 where additional continuity equations are advanced in the prediction step. This leads to the following update (here given for u)

$$\mathcal{L}_{\text{conv}} = \frac{1}{\rho_u^*}\nabla_f \cdot (\rho_u^n u^n \mathbf{u}^n) - \frac{u^n}{\Delta t}\frac{\rho_u^* - \rho_u^n}{\rho_u^*} \quad . \tag{6.8}$$

The additional term in the RHS is a correction of momentum due to the mass change in the cell during the prediction step. The predicted staggered density ρ_u^* is obtained from the following continuity equation

$$\frac{\rho_u^* - \rho_u^n}{\Delta t} = -\nabla_f \cdot (\rho_u^n \mathbf{u}^n) \quad . \tag{6.9}$$

Note that ρ_u^* is only an intermediate evolution variable discarded just after prediction step. In the general case, ρ_u^* will differ from the reconstruction ρ_u^{n+1}. The staggered density at t^n is obtained from the following equation

$$\rho_{u,i-\frac{1}{2}}^n = \rho_g + \theta_{u,i-\frac{1}{2}} [\rho]_\Gamma \quad , \tag{6.10}$$

with $\theta_{u,i-\frac{1}{2}}$ a staggered volume fraction approximation.
For VOF and CLSVOF, the PLIC reconstruction is available and is used to reconstruct $\theta_{i-\frac{1}{2}}$ as proposed in [159].
When using SLS, the distance function ϕ is directly used such that

$$\theta_{u,i-\frac{1}{2}} = \begin{cases} 0 & \text{if } \phi_{i-1}^n < 0 \text{ and } \phi_i^n < 0 \\ 1 & \text{if } \phi_{i-1}^n > 0 \text{ and } \phi_i^n > 0 \\ \frac{\max(0,\phi_{i-1}^n)+\max(0,\phi_i^n)}{|\phi_{i-1}^n|+|\phi_i^n|} & \text{otherwise} \end{cases} \quad . \tag{6.11}$$

In this work, it was found that using the regularized Heaviside H_Γ^ϵ as in [81] was a better choice for momentum conservation and solver robustness in the case of ACLS. Indeed, Eq. (6.11) relies on geometric informations provided by a distance function. Because of the errors in ϕ_{FMM}, this expression can lead to huge errors in density. Hence, a more robust approximation of ρ_u^n is performed by using the simple ψ average

$$\theta_{u,i-\frac{1}{2}} = \frac{1}{2}(\psi_{i-1} + \psi_i) \quad . \tag{6.12}$$

Finally, the fluxes $F^{(\rho)}$ and $F^{(\rho u)}$ have to be evaluated with the same interpolation for consistency. Moreover, the scheme have to be bounded to avoid any overshoot or undershoot of density as discussed in Section 5.2.5.3. In the present solver, a WENO5 interpolation is performed, which switches to an upwind evaluation when the stencil crosses the interface.

6.1.1.2 Capillary term $\mathcal{L}_{\text{cap}}$

As discussed in Section 5.4.1, it is not possible to represent the Heaviside function at the exact interface location, and a numerical approximation is required leading to the following definition of $\mathbf{F}_\sigma$

$$\mathbf{F}_\sigma = \sigma\kappa\nabla H_\Gamma^{num} \tag{6.13}$$

with H_Γ^{num} a discrete representation of the Heaviside.
A CSF approach would use a smooth Heaviside $H_\Gamma^{num} = H_\Gamma^\epsilon$. The expression corresponding to GFM is the sharp Heaviside defined at the cell center $H_\Gamma^{num} = H_\Gamma^0 = H_\Gamma$.
The GFM discretization of $\mathcal{L}_{\text{cap}}$ used in our solver can be written (here for u) as

$$\mathcal{L}_{\text{cap}} = \frac{1}{\rho_u^{n+1}}\sigma\kappa_u^{n+1}\nabla_f^x H_\Gamma^{0,n+1} \quad . \tag{6.14}$$

From the discussion in Section 5.4.3, the following choices have been made for curvature computation :

- For SLS, the curvature is retrieved by using second-order finite differences (FD) on ϕ while the Least-Square minimization (LSQUAD) of [38] is used on ϕ_{FMM} and ϕ_{PLIC} for ACLS and CLSVOF respectively. The curvature is then defined at the cell center and the harmonic interpolation Eq. (5.70) is used to get face curvatures. For robustness improvement, the interpolation switches to linear interpolation Eq. (5.69) when curvature changes sign as in [38].
- For VOF, the curvature is computed from the general height function (HF) method [169] and interpolated to the face using Eq. (5.74).

6.1.1.3 Viscous term $\mathcal{L}_{\text{visc}}$

Following the conclusions given in Section 5.3.4, an explicit WDF is used here for its simplicity and good results. The viscous term $\mathcal{L}_{\text{visc}} = \frac{1}{\rho^{n+1}}\nabla_f \cdot \left(\mu^{n+1}\mathbf{D}^n\right)$ is discretized as (here for u)

$$\begin{aligned}\mathcal{L}_{\text{visc}} = \frac{1}{\rho_u^{n+1}} \big(&2\nabla_f^x \left(\mu^{n+1}\nabla_c^x u^n\right) \\ &+ \nabla_c^y \left(\mu^{n+1}\nabla_f^y u^n\right) \\ &+ \nabla_c^y \left(\mu^{n+1}\nabla_f^x v^n\right)\big) \quad .\end{aligned} \tag{6.15}$$

In this expression, some μ^{n+1} values are needed at Σ_ρ corners, they are defined with simple average of the neighbour collocated values

$$\mu_{i-\frac{1}{2},j-\frac{1}{2}}^{n+1} = \frac{1}{4}\left(\mu_{i,j}^{n+1} + \mu_{i-1,j}^{n+1} + \mu_{i,j-1}^{n+1} + \mu_{i-1,j-1}^{n+1}\right) \quad . \tag{6.16}$$

Note that sharper choices of corner interpolation can be made based on a distance function [212] but no significant improvements have been observed in the test cases presented in this work.

6.1.2 Correction step

The correction step is written as

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{*}}{\Delta t}=-\frac{1}{\rho_f^{n+1}}\nabla_f P^{n+1} \quad , \tag{6.17}$$

where the pressure P^{n+1} is obtained from the resolution of variable coefficient Poisson equation (here using the linear solver library of PETSc[14])

$$\nabla_c\cdot\left(\frac{1}{\rho_f^{n+1}}\nabla_f P^{n+1}\right)=\frac{1}{\Delta t}\nabla_c\cdot\mathbf{u}^{*} \quad , \tag{6.18}$$

with $\rho_f^{n+1}=\rho_u^{n+1}$ for the x-normal faces (ρ_v^{n+1} and ρ_w^{n+1} for y and z-normal faces).

6.1.3 Summary of the two-phase solver procedure

A summary of the adaptations introduced to unify all Eulerian methods in the same low Mach solver is provided in Table 6.1 and the full algorithm for a timestep is given:

1. Compute ρ_f^n from c^n;
2. Advance interface following one of the algorithm of Section 4.1 to obtain c^{n+1};
3. Compute κ_f^{n+1}, $H_\Gamma^{0,n+1}$, μ^{n+1}, ρ_f^{n+1} from c^{n+1};
4. Set $\mathbf{u}^k=\mathbf{u}^n$ and loop in a RK2-SSP time integration from $k=1$ to $k=3$
   (i) Advance $\mathbf{u}^n$ and ρ_f^n to ρ_f^* and $\mathbf{u}^*$ with the consistent mass and momentum scheme of Section 6.1.1 and fluxes computed using $\mathbf{u}^k$;
   (ii) Solve the Pressure from Eq. (6.18);
   (iii) Correct the velocity to obtain $\mathbf{u}^{k+1}$ with Eq. (6.17);

The timestep has a stability constraint based on the CFL, the surface tension and the viscosity :

$$\Delta t<\min\left(\frac{\Delta x}{2\|\mathbf{u}\|},\sqrt{\frac{\Delta x^3(\rho_l+\rho_g)}{(2\pi)^3\sigma}},\frac{\Delta x^2}{4\max(\nu_l,\nu_g)}\right) \quad . \tag{6.19}$$

In most of the application shown hereafter, the surface tension restriction is dominant.

method	VOF	SLS	ACLS	CLSVOF
κ	HF	FD	LSQUAD	LSQUAD
κ_f	Eq. (5.74)	Eq. (5.70)	Eq. (5.70)	Eq. (5.70)
H_Γ^ϵ	f	Eq. (2.37)	ψ	f
θ_f	From PLIC	Eq. (6.11)	Eq. (6.12)	From PLIC

Table 6.1: *Summary of computation choices for the four interface capturing methods*

6.2 Numerical results

6.2.1 Surface tension modelling

This section aims to assess the surface tension modelling implying both curvature computation and surface tension force discretization. In a first test, the curvature computation accuracy is evaluated without taking into account the transport errors induced by the velocity field. When an interface capturing method is coupled with a two-phase solver, the curvature errors act as a vorticity source in the momentum equation scaling with $\nabla\kappa$ and produce well-known parasitic currents [1]. To enlighten this behaviour, two additional cases are considered and illustrated in Fig.6.1: a static test case where only curvature error acts as a source of error and a dynamic case where the interface transport acts as a second source of error.
Finally, the canonical damping wave test case is considered to assess the solver on capillary-diven flows.

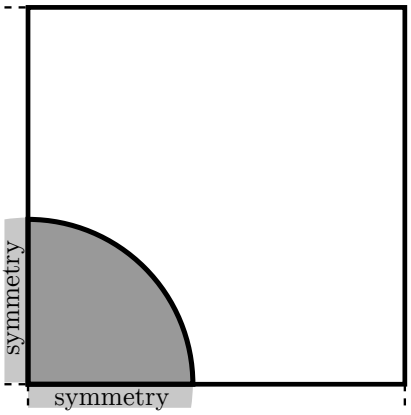
6.2.1.1 Curvature computation

First, the accuracy of the curvature is demonstrated for the different representations of the interface. The choices of κ computation summed up in Tab.6.1 are considered. Note that for ACLS, ϕ_{FMM} is used while for CLSVOF ϕ_{PLIC} is used instead of the exact distance function ϕ.
The relative curvature errors are defined as in [38] :

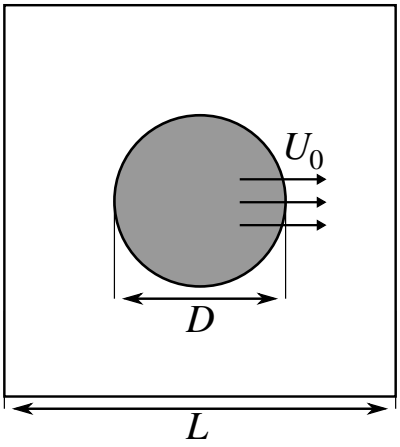
$$L_2(\kappa) = \frac{\sqrt{\frac{1}{N_\Gamma}\sum_{i=1}^{N_\Gamma}\left(\kappa_{exact} - \kappa_{f,i}\right)^2}}{\kappa_{exact}} \quad , \tag{6.20}$$

$$L_\infty(\kappa) = \frac{\max\limits_i \left|\kappa_{exact} - \kappa_{f,i}\right|}{\kappa_{exact}} \quad , \tag{6.21}$$

with κ_{exact} the exact curvature and κ_f an interpolated curvature to the faces where $\nabla_f H_\Gamma^0$ is non-zero. In practice, the values of κ_f are the only curvature values used for surface tension modelling regardless of the interface capturing method.
For a range of cells per diameter $N_D = 3.2$ to $N_D = 409.6$, the errors are evaluated on 100 circles randomly located in the domain in order to meet as much configurations as possible. The final metric is $\langle L_2(\kappa)\rangle$ the mean of $L_2(\kappa)$ and $\max\left(L_\infty(\kappa)\right)$ the maximum of $L_\infty(\kappa)$ over all these configurations.
Figure 6.3a and Fig. 6.3b show convergence in L_2 and L_∞ for all methods but ACLS. The ACLS method shows a saturation of the error convergence for high resolutions because of the second-order nature of the ϕ_{FMM} distance function. Note that for completeness, Fig. 6.4a and Fig. 6.4b illustrate the difference between LSQUAD and FD approach applied on ϕ_{FMM}. It can be seen that the FD method does not show any convergent behaviour while LSQUAD, the method used here for CLSVOF and ACLS, manages to decrease the error until $N_D = 51.2$.
This saturation is not observed for the CLSVOF approach because of the relaxation applied on the ϕ_{PLIC}. This takes the form of $\phi_{PLIC} = \omega\phi + (1-\omega)d$ and ω is a function of the



(a) *Static test case*



(b) *Dynamic test case*

Figure 6.1: *Test case set up for the spurious currents quantification*

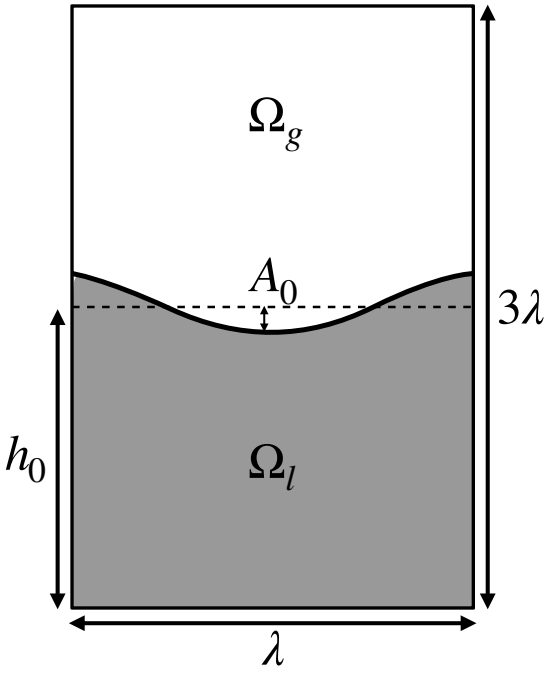
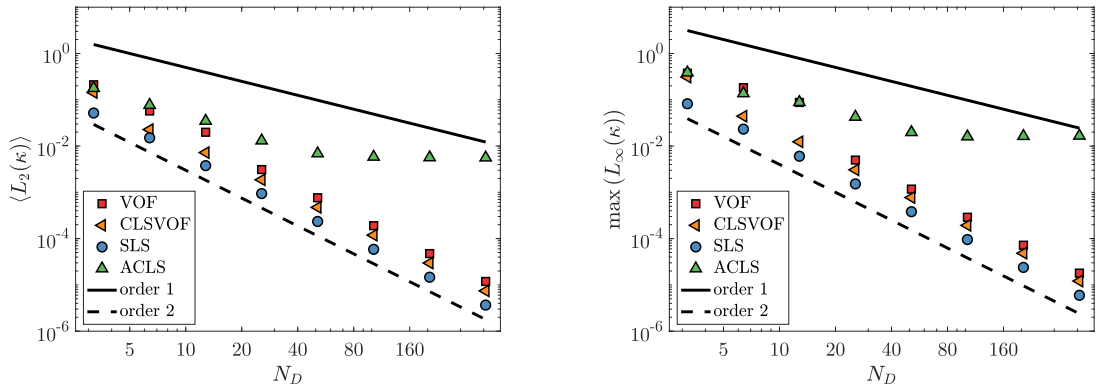


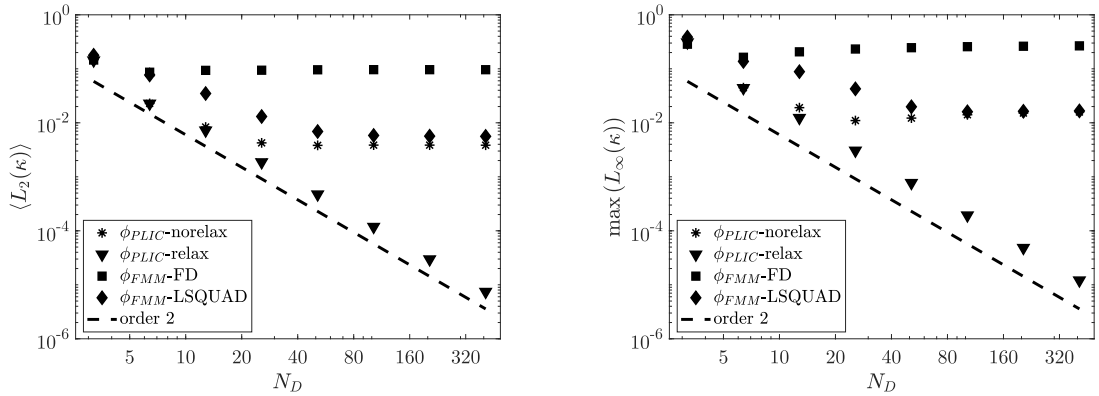
Figure 6.2: *Planar damping wave simulation set up*

difference between ϕ and d. As observed in Fig. 6.4a and Fig. 6.4b, if no relaxation is performed, the curvature computation does not converge at high resolution for the same reasons as for ACLS. If the relaxation is activated, ω will tend to one for high resolution as the local curvature in a given cell is closer to a plane, so the discrepancy between ϕ and d is lower. This allows CLSVOF curvature to show convergence even for high resolutions. Finally, HF method has a transient convergence regime in the lowest resolutions because of the parabola fit switch as already pointed out in [169].



(a) *Mean of* $L_2(\kappa)$ **(b)** *Maximum of* $L_\infty(\kappa)$

Figure 6.3: *Mesh convergence of* $\langle L_2(\kappa)\rangle$ *and* $\max(L_\infty(\kappa))$ *for the 4 methods*



(a) *Mean of* $L_2(\kappa)$ **(b)** *Maximum of* $L_\infty(\kappa)$

Figure 6.4: *Mesh convergence of* $L_2(\kappa)$ *and* $L_\infty(\kappa)$ *error for other choices of* κ *computation*

6.2.1.2 Spurious currents analysis

The static droplet is a good test case to assess a surface tension model. A complete investigation has been done by Abadie et al. in [1] where they compare various $\mathbf{f}_\sigma$ discretizations and κ computation in both VOF and SLS framework.
When a well-balanced discretization is used, the parasitic currents only depends on κ and H_Γ. Following [1], taking the curl of Eq. (5.53) leads to a new requirement for the Laplace equilibrium

$$\nabla\kappa \times \nabla H_\Gamma = 0 \quad . \tag{6.22}$$

It is also possible to write the vorticity transport equation from Eq. (2.26) leading to

$$\frac{\partial \boldsymbol{\omega}}{\partial t} + (\mathbf{u}\cdot\nabla)\boldsymbol{\omega} + (\boldsymbol{\omega}\cdot\nabla)\mathbf{u} = \nu\Delta\boldsymbol{\omega} - \frac{\sigma}{\rho}\nabla\kappa \times \nabla H_\Gamma \quad . \tag{6.23}$$

It appears clearly that an error in curvature will lead to a source term in vorticity.
In a viscosity dominant flow, considering a long enough time t_ν, the viscous contribution will balance the vorticity source term to reach a steady state

$$\nu\Delta\boldsymbol{\omega} = \frac{\sigma}{\rho}\nabla\kappa \times \nabla H_\Gamma \quad . \tag{6.24}$$

By dimensional consideration, one can argue that $\nu\Delta\boldsymbol{\omega} \sim \nu U_\sigma / D^3$ and $\sigma\nabla\kappa\times\nabla H_\Gamma \sim \sigma/D^3$ with U_σ a spurious velocity and D the droplet diameter. Thus, a characteristic spurious velocity related to viscous flows can be determined

$$U_{\sigma,\nu} = \frac{\sigma}{\rho\nu} \quad . \tag{6.25}$$

In an inertia dominant flow, the vorticity source term is balanced by the first term of Eq. (6.23)

$$(\mathbf{u}\cdot\nabla)\boldsymbol{\omega} = \frac{\sigma}{\rho}\nabla\kappa \times \nabla H_\Gamma \quad . \tag{6.26}$$

By dimensional consideration, $(\mathbf{u}\cdot\nabla)\boldsymbol{\omega} \sim \rho U_\sigma^2 / D^2$ and a characteristic spurious velocity linked to inertial flows is defined as

$$U_{\sigma,\rho u} = \sqrt{\frac{\sigma}{\rho D}} \quad . \tag{6.27}$$

The time scale related to viscosity can then be defined as

$$t_\nu = \frac{D^2}{\nu} \quad , \tag{6.28}$$

and the time scale related to capillarity effects

$$t_\sigma = \sqrt{\frac{\rho D^3}{\sigma}} \quad . \tag{6.29}$$

From these two characteristic quantities, one can define the Capillarity number Ca such that

$$Ca = \frac{\rho\nu U_{\sigma,\nu}}{\sigma} \quad , \tag{6.30}$$

and a Laplace number La

$$La = \left(\frac{U_{\sigma,\nu}}{U_{\sigma,\rho u}}\right)^2 = \left(\frac{t_\nu}{t_\sigma}\right)^2 = \frac{\sigma D}{\rho\nu^2} \quad . \tag{6.31}$$

Now that the source of spurious currents related to surface tension has been identified, different test cases are performed to describe the behaviour of the different interface capturing methods on such problematic.

6.2.1.3 Static test case

An infinite cylinder of diameter $D = 0.4$ is centered in a $[1 \times 1]$ domain without gravity force. Only a quarter of the domain is considered here where right and top boundary conditions are no-slip walls and bottom and left boundary conditions are symmetry. The cylinder is supposed to stay at rest as the pressure force is expected to balance exactly the capillary forces.

Exact curvature The curvature κ is imposed constant with the exact value $2/D$. The pressure jump is then known analytically as $\Delta P = \sigma\kappa = 2\sigma/D$. The test case is performed for 3 different Heaviside formulations all written in the well-balanced way : $H^{\epsilon}_{\Gamma}(f)$, $H^{\epsilon}_{\Gamma}(\psi)$ and H^{0}_{Γ} on a grid $N_x = 51$ in both directions. This choice of mesh size is made to have the centreline in the y-direction matching exactly the location of the cell centers at $i = 26$. This part aims to show the different pressure jump resolutions with respect to the choice of H_Γ in the $\mathbf{f}_\sigma$ computation. Note that $H^{\epsilon}_{\Gamma}(\psi) = H^{\epsilon}_{\Gamma}(\phi)$ so the resolution is the same for smooth Heaviside from SLS or ACLS framework.

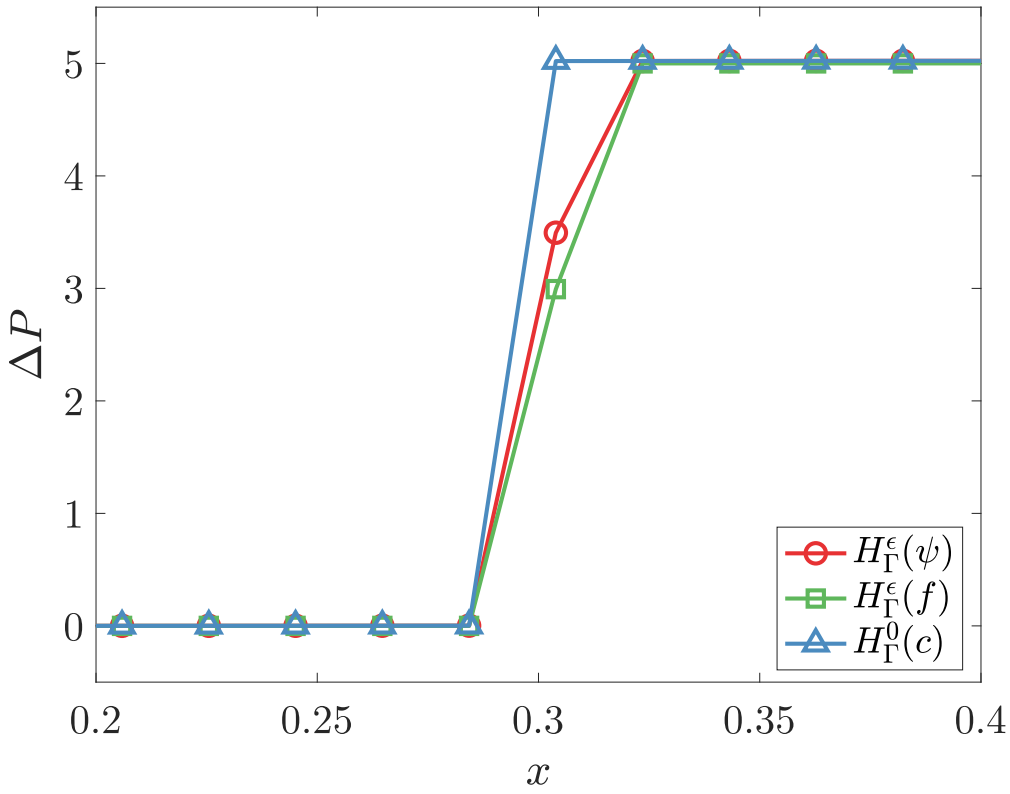
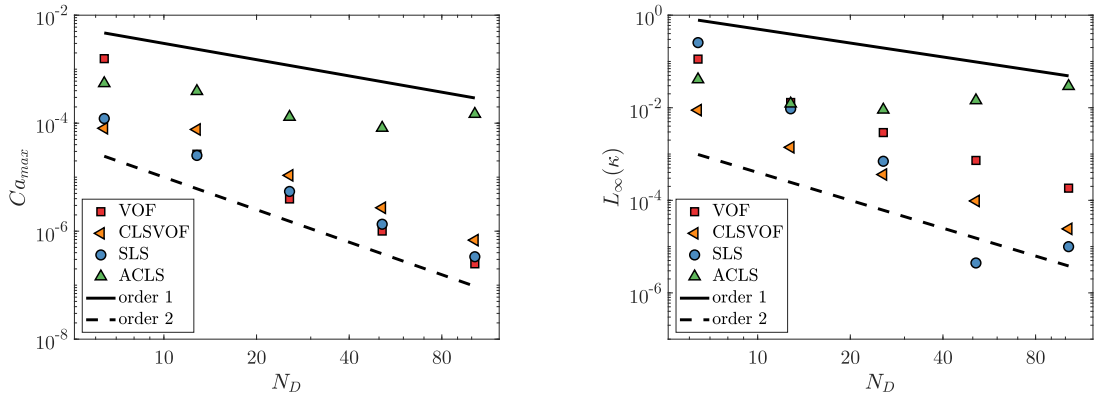


Figure 6.5: *Pressure resolution for different H_Γ choices*

The pressure profile on the centreline in the y-direction is showed in Fig. 6.5. One can

notice that the pressure matches perfectly with the Heaviside profile as expected in a well-balanced method (in light of Eq. (5.55)). Moreover, velocity is at zero machine in all three cases.

Computed curvature For the same configuration but with a computed curvature, the methods are not able to compute a constant κ and spurious currents are present in the domain. This behaviour is quantified here with the maximum Capillary number Ca_{max}. The fluid properties are the same in both phases such that $La = 12000$.

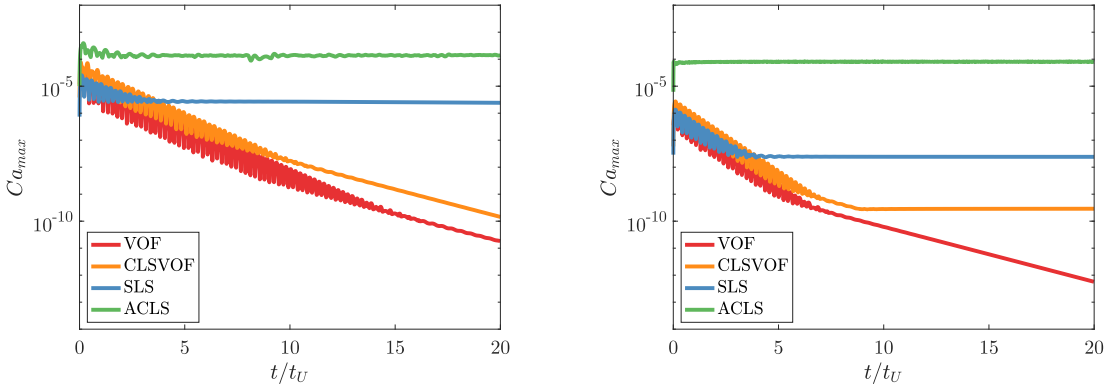


(a) *Ca_{max} for a full run* **(b)** *$L_\infty(\kappa)$ error at $t/t_\sigma = 20$*

Figure 6.6: *Mesh convergence of Ca_{max} and $L_\infty(\kappa)$ error for the static case*

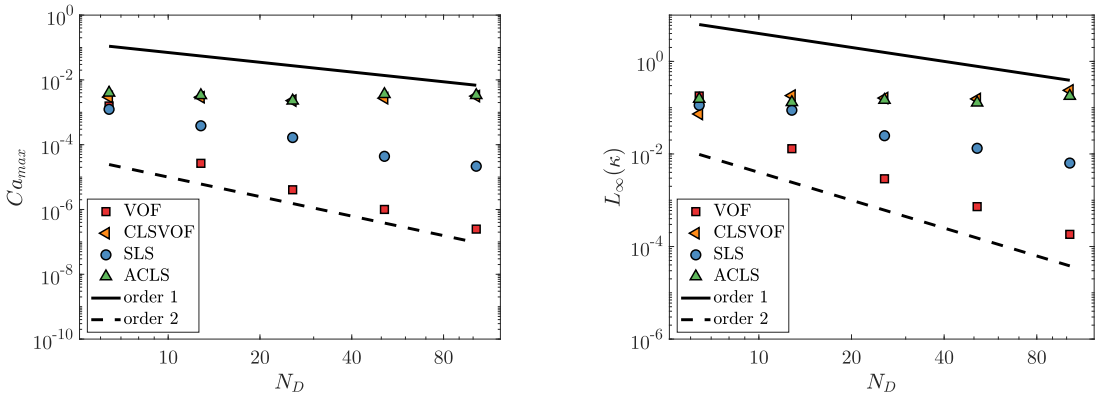
In Fig. 6.6a is displayed the mesh convergence of the maximum Ca_{max} during a static simulation. All methods but ACLS converge with mesh resolution. This is because of the non-converging behaviour of κ which has been demonstrated in Section 6.2.1.1 and from the $L_\infty(\kappa)$ error at the end of the simulation in Fig. 6.6b. This has been previously observed in [38] in 3D where the Ca_{max} was closely the same for the 40^3 and 80^3 meshes.
The damping of the spurious currents in Fig. 6.7 can be explained by the numerical curvature computation. The initial spurious currents are a direct consequence of the initial error introduced by the curvature computation. VOF is able to reach an equilibrium close to zero machine as shown in [169]. The other methods also reach a steady Ca_{max} value which is not zero because of the reinitialization step introducing new errors at each iteration [1]. While this is very noticeable for ACLS, the magnitude decreases using SLS and decreases even more with CLSVOF.

Effect of smoothing the Heaviside For the same configuration another choice of Heaviside is done for the methods. Instead of taking the GFM with the sharp Heaviside H^0_Γ, the CSF approach is taken with the smooth heaviside H^ϵ_Γ summarized in Table 6.1. The aim is to observe the impact of smoothing the Heaviside on the spurious currents.
In Fig. 6.8a is displayed the mesh convergence of the maximum Ca_{max} with the smooth Heaviside H^ϵ_Γ. One can see that VOF and SLS still converge at second-order and first-order respectively and ACLS and CLSVOF do not converge anymore. The evolution of



(a) $N_D = 12.8$ **(b)** $N_D = 51.2$

Figure 6.7: *Temporal evolution of Ca_{max} for the static case*

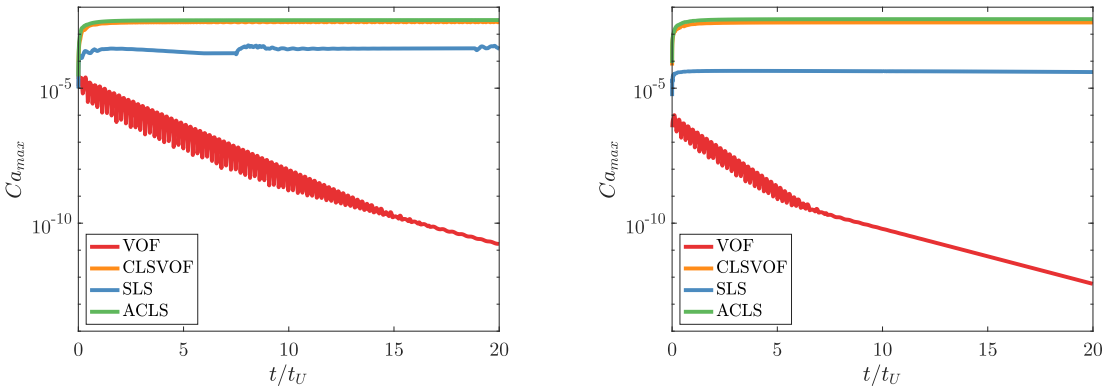


(a) Ca_{max} *for a full run* **(b)** $L_\infty(\kappa)$ *error at* $t/t_\sigma = 20$

Figure 6.8: *Mesh convergence of Ca_{max} and $L_\infty(\kappa)$ error for the static case with H^ϵ_Γ*

maximum Capillary number Ca_{max} is presented in Fig. 6.9. The VOF method is able to damp the spurious currents up to zero machine even with a smoother Heaviside while all other methods fail to damp the spurious currents. However, SLS is able to keep them at a fairly low magnitude, which is an acceptable result compared to ACLS and CLSVOF. Finally, comparing Fig. 6.8a and Fig. 6.6a, a difference in magnitude between H^ϵ_Γ and H^0_Γ for CLSVOF, SLS and ACLS can be observed while it has not a significant impact on the result using VOF.

The explanation of such differences in the results for different heaviside definitions reside in the curvature computation. In VOF, the curvature is computed from height functions, so it is located at the interface. Then, even for a smoother definition of the Heaviside, the face interpolations of κ are made from good estimates κ_Γ and do not add additional errors. For other methods, the curvature estimation is located at cell centers. It has been shown that a pretty good estimation of κ_Γ can be obtained from either linear Eq. (5.69)



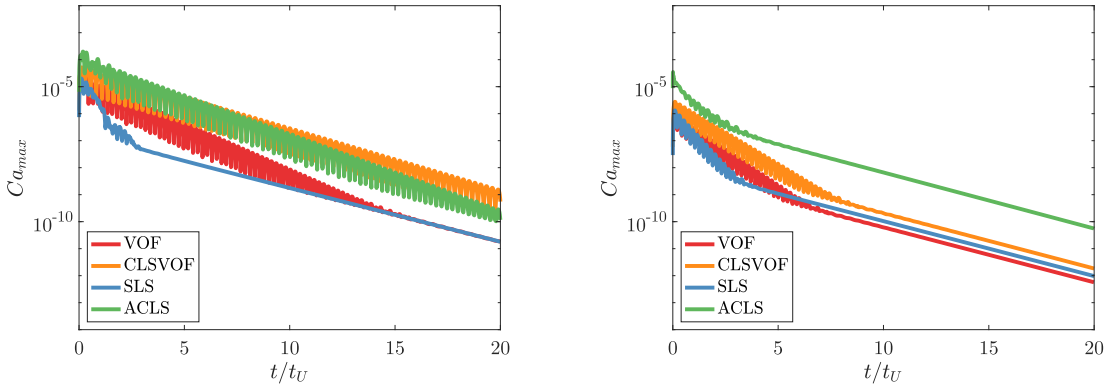
(a) $N_D = 12.8$ **(b)** $N_D = 51.2$

Figure 6.9: *Temporal evolution of Ca_{max} for the static case with H^{ϵ}_{Γ}*

or harmonic interpolations Eq. (5.70) when the interface lies between the cell centers (case of $\kappa_{i,j+\frac{1}{2}}$ in Fig. 5.9b). However, when the face curvature is computed from two values of a same fluid phase, the estimation is very bad because of the use of the average Eq. (5.71) of two already erroneous κ values (case of $\kappa_{i,j-\frac{1}{2}}$ in Fig. 5.9b). This results in a first-order error on the curvature. When using H^0_{Γ}, this is not a problem as only face curvature of cell centers belonging to different phases will be used. This problem can be avoided by using a height function approach for curvature computation as demonstrated in [1].
Finally the ACLS and CLSVOF approaches seem to have higher spurious currents than SLS. This is also due to the curvature error which is higher for methods using a signed distance more prompt to numerical errors to compute curvature.

Effect of reinitialization in the Level Set framework In Fig. 6.10, test cases without reinitialization are included. If no reinitialization is performed, CLSVOF, SLS and ACLS are able to retrieve the equilibrium observed in the VOF context. The reinitialization step acts as an additional source of errors in the curvature as it creates small displacements of the zero-isocontour. Then the curvature computation does not depends only on the velocity field but also on the reinitialization step. This explains why the viscosity contribution cannot balance the vorticity source in Eq. (6.23).

Laplace number effect on the result Finally, the same test case is performed with a fixed grid size $N_D = 12.8$.
The goal here is to observe the sensitivity of the spurious currents to the Laplace number and give an understanding of the spurious currents using the analysis of Section 6.2.1.2.
Only the VOF method is considered here to reduce the number of simulations to perform. Figure 6.11 shows the results of the different La. For all La, the method is able to damp oscillations up to zero machine in a time $t\nu$. This is expected regarding the above dimensionless analysis : at t_ν, the viscous contribution exactly balances the vorticity source term



(a) $N_D = 12.8$ **(b)** $N_D = 51.2$

Figure 6.10: *Temporal evolution of Ca_{max} for the static case without reinitialization*

of Eq. (6.23).

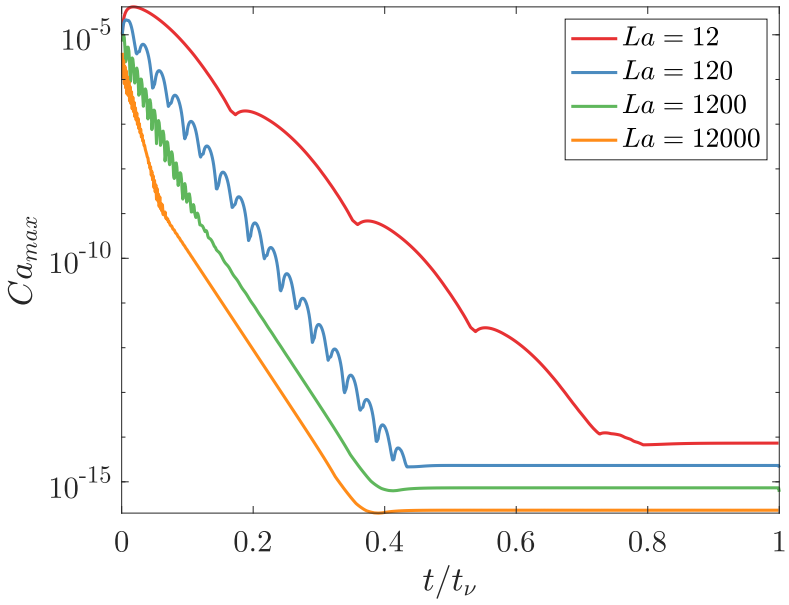
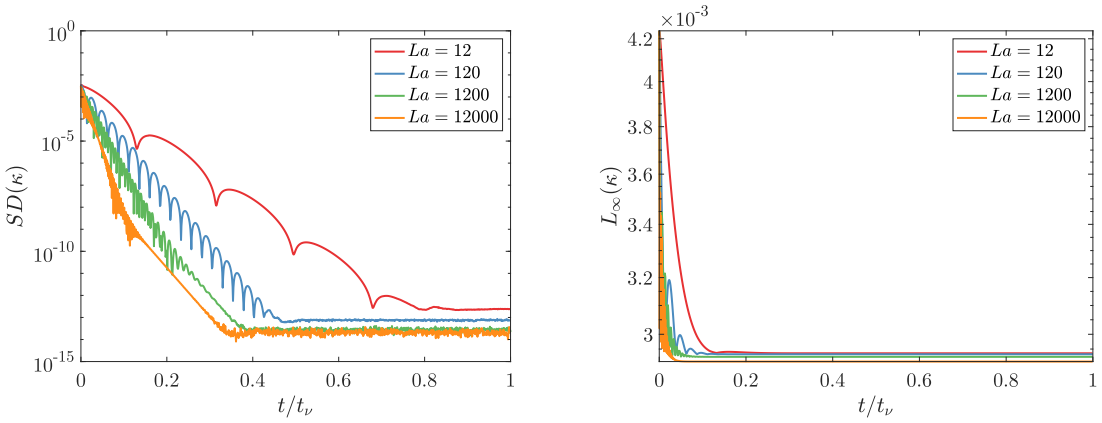


Figure 6.11: *Comparison of the Ca_{max} evolution in time for different La using the VOF algorithm*

The damping of the spurious current can be explained by the numerical curvature computation. The initial spurious currents are a direct consequence of the initial error introduced by the curvature computation. It is very important to notice that the important quantity is not the curvature error itself but the standard deviation as the vorticity source comes from a gradient of curvature $\nabla\kappa$.
For a deeper comprehension of this phenomenon, the standard deviation of curvature $SD(\kappa)$



(a) *Standard deviation of curvature* **(b)** *Maximum curvature error*

Figure 6.12: *Standard deviation and maximum error of curvature for different La using the VOF algorithm*

and the maximum relative error in curvature $L_\infty(\kappa)$ are introduced

$$SD(\kappa) = \sqrt{\frac{1}{N_\Gamma}\sum_{i}^{N_\Gamma}(\bar{\kappa}_{num} - \kappa_{num,i})^2} \quad , \tag{6.32}$$

$$L_\infty(\kappa) = \frac{\max_i |\kappa_{ex} - \kappa_{num,i}|}{\kappa_{ex}} \quad , \tag{6.33}$$

with $\bar{\kappa}_{num}$ the mean numerical curvature and N_Γ the number of cells where the curvature is computed (and used for surface tension force computation).
In Fig. 6.12a, the standard deviation of the computed κ is presented with respect to time. It appears clearly that the reduction of the spurious currents are related to the reduction of the standard deviation of κ. The moment when spurious currents are up to zero machine is the moment when the droplet have reached its numerical equilibrium with a constant κ_{num} in all the interface cells. Note that κ_{num} is different from κ_{exact} has shown in Fig. 6.12b : the error is about 0.1%.
Finally, the same test case has been done for an inviscid droplet. The results are shown in Fig. 6.13 where the time and velocity scales are taken as t_σ and $U_{\sigma,\rho u}$ respectively. In this zoomed figure, it is clear that all spurious currents are damped excepted the inviscid case. This can be explained by the above investigation. In such case, there are no dissipation process able to damp the initial disequilibrium induced by curvature computation. The droplet will then oscillate around its numerical equilibrium without any damping effect. Following this analogous description of the test case, Popinet defined in [169] an estimate of the oscillation wavelength λ for an infinite cylinder prone to small amplitude perturbations. The period can be defined as t_λ which can be bluntly read on Fig. 6.13 as $0.4t_\sigma$:

$$t_\lambda = \sqrt{\frac{\rho\lambda^3}{\pi\sigma}} \approx 0.4t_\sigma \quad , \tag{6.34}$$

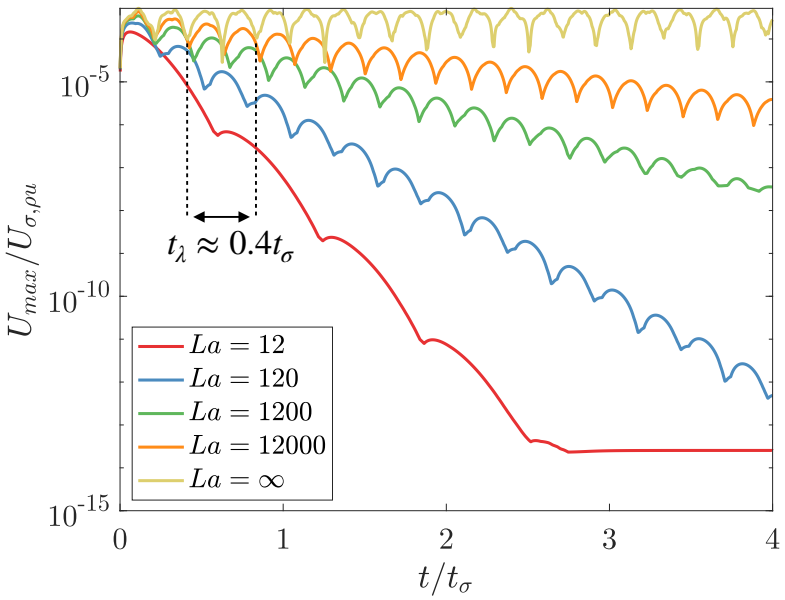


Figure 6.13: *Comparison of U_{max} evolution in time for different La*

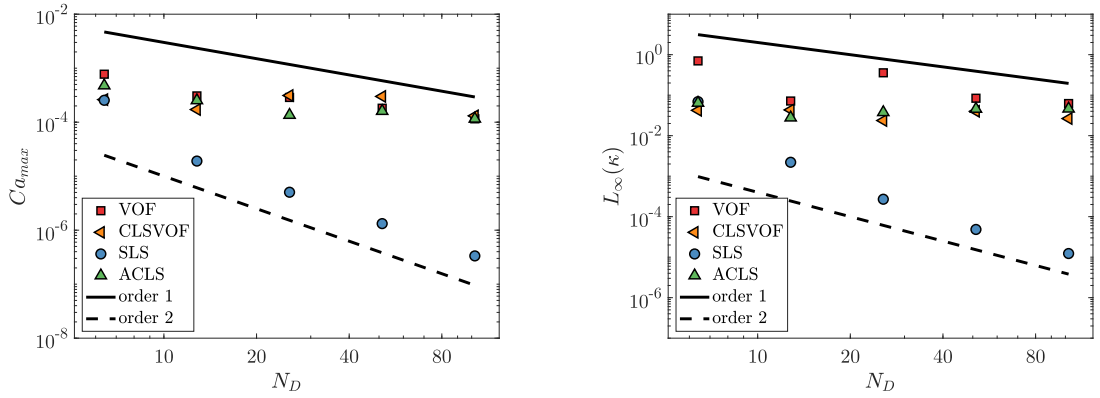
which leads to a wavelength

$$\lambda \approx 0.8D \approx \frac{\pi D}{4} \quad , \tag{6.35}$$

which corresponds to the perimeter of our droplet quarter.
The only difference in the results of Popinet in [169] is that the inviscid case was damped with time. This can be explained by the numerical scheme used for convection : Popinet is using a second-order approach which is more numerically diffusive than the WENO5-RK2 scheme used in the present investigation. The damping of his inviscid droplet is then a consequence of the small numerical diffusion of his convection scheme.

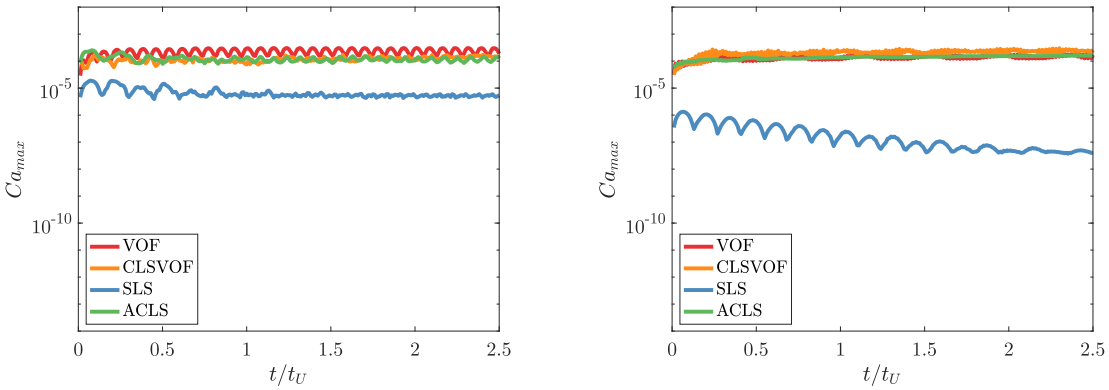
6.2.1.4 Dynamic test case

In order to quantify the impact of the flow dynamic on the spurious currents, the following test case is considered. An infinite cylinder of diameter $D = 0.4$ is centered in a $[1 \times 1]$ domain with a uniform horizontal velocity U_0 where boundary conditions are periodic in the velocity direction and free slip conditions are imposed on the top and bottom boundaries. From the new velocity scale U_0, a new time scale can be defined as $t_U = \frac{D}{U_0}$ and the Weber number $We = \frac{\rho U_0^2 D}{\sigma}$. The fluid properties are the same as in the static case with $We = 0.4$. The mesh convergences of Ca_{max} and $L_\infty(\kappa)$ after one revolution are illustrated in Fig. 6.14a and 6.14b. As in [169] and [1], no convergence is observed for VOF because of the second-order errors introduced by the transport step. This is also true for CLSVOF and ACLS. However, SLS exhibits a huge reduction of spurious currents explained by the higher accuracy of curvature and transport.



(a) Ca_{max} *for a full run* **(b)** $L_\infty(\kappa)$ *error at* $t/t_U = 2.5$

Figure 6.14: *Mesh convergence of* Ca_{max} *and* $L_\infty(\kappa)$ *error for the dynamic case*

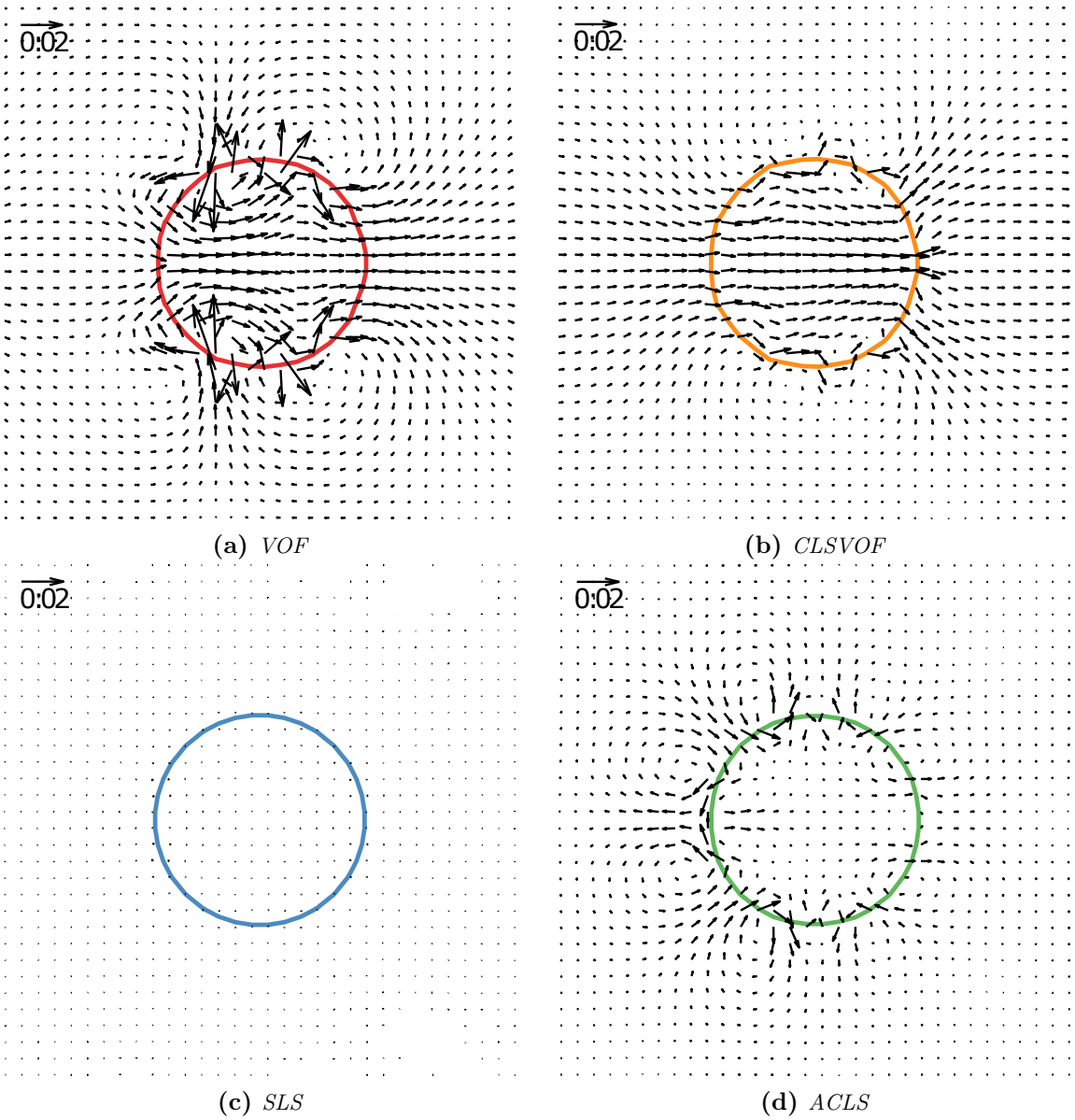


(a) $N_D = 12.8$ **(b)** $N_D = 51.2$

Figure 6.15: *Temporal evolution of* Ca_{max} *for the dynamic case*

Compared to the static case, no damping of spurious currents is observed in Fig. 6.15. This observation puts in evidence the inability of numerical methods to retrieve an exact balance between pressure and surface tension in a dynamic test case. The permanent introduction of transport errors acts as an imbalance in the curvature computation. This shows the tight coupling between the advection scheme and the surface tension contribution in a two-phase solver.

For a more complete visualization, the velocity field in the reference of the translating droplet is given for a medium resolution of $N_D = 12.8$ in Fig. 6.16. CLSVOF is slightly improving VOF curvature computation but it is not as accurate as SLS because of the ϕ_{PLIC} perturbations introduced by the LS-VOF coupling. ACLS performs better than VOF and CLSVOF in the medium resolution range while SLS maintains a fairly low amount of spurious currents. The spurious intensity $\|\mathbf{u}\|_{max}$ represents about 1% of U_0 for VOF, CLSVOF and ACLS against 0.001% for SLS.



(a) *VOF* **(b)** *CLSVOF*

(c) *SLS* **(d)** *ACLS*

Figure 6.16: *Relative velocity field (U_0 is subtracted for visualization) for the dynamic case at* $N_D = 12.8$

6.2.1.5 Planar damping wave

The planar damping wave is an interesting test case as an analytical solution is available in the literature. The test case has been widely investigated as a solver validation[172; 73; 65; 49; 169; 66; 90]. Here, a planar wave is initialized with a small harmonic perturbation of amplitude A_0 and both fluids are at rest with the same density and viscosity properties.

The initial interface height can be described by

$$y_0 = h_0 + A_0 \cos\left(\frac{2\pi x}{\lambda}\right) \quad , \tag{6.36}$$

with λ the wavelength of the perturbation and $h_0 = 3\lambda/2$ the vertical interface position. λ is taken to unity and $A_0 = \lambda/100$. In this problem, non dimensional time and viscosity are defined as

$$\tau = t\omega_0, \quad \xi = \nu\lambda^2/\omega_0 \quad , \tag{6.37}$$

with the perturbation frequency $\omega_0 = \sqrt{\sigma\lambda^3/(\rho_l + \rho_g)}$.
The perturbation amplitude is then deduced from the analytical solution derived by Prosperetti et al. [173]

$$\begin{aligned} A(\tau) =& A_0 \frac{4(1-4\beta)\xi^2}{8(1-4\beta)\xi^2 + 1} \operatorname{erfc}\left(\sqrt{\xi\tau}\right) \\ &+ \sum_{i=1}^{4} A_0 \frac{z_i}{Z_i} \frac{\omega_0^2}{z_i^2 - \xi\omega_0} \exp\left(\frac{\left(z_i^2 - \xi\omega_0\right)\tau}{\omega_0}\right) \operatorname{erfc}\left(z_i\sqrt{\frac{\tau}{\omega_0}}\right) \quad , \end{aligned} \tag{6.38}$$

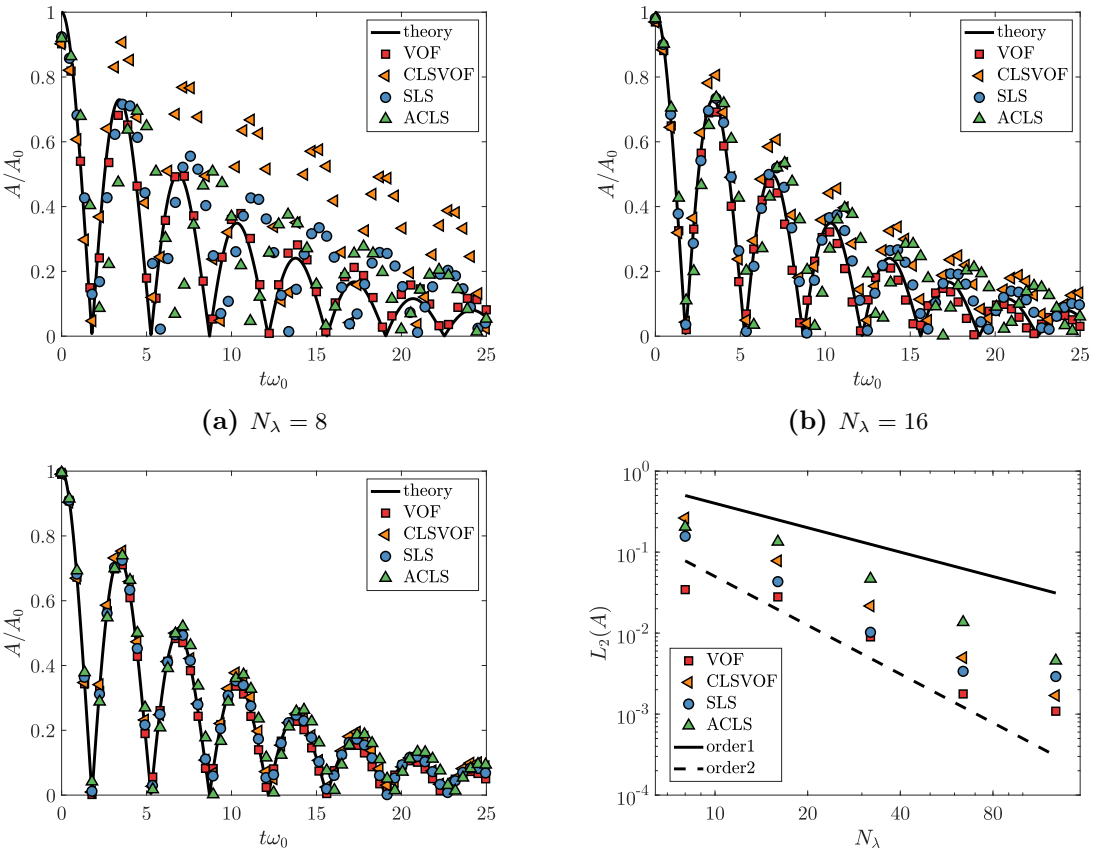
with $\beta = \rho_l\rho_g/(\rho_l + \rho_g)^2$, $Z_i = \prod_{\forall j \neq i} (z_j - z_i)$ and z_i the four complex roots of the following quartic equation in z

$$\begin{aligned} z^4 - 4\beta\left(\xi\omega_0\right)^{1/2} z^3 + 2(1-6\beta)\xi\omega_0 z^2 \\ + 4(1-3\beta)\left(\xi\omega_0\right)^{3/2} z + (1-4\beta)\left(\xi\omega_0\right)^2 + \omega_0^2 = 0 \quad . \end{aligned} \tag{6.39}$$

In the special case of same density, momentum and viscosity jumps at the interface cancel and the numerical errors are only due to curvature computation and interface transport. The densities are $\rho_l = \rho_g = 1$ which leads to $La = 3000$, $\xi = 0.0647$ and $\beta = 0.25$. The solution holds for infinite domain in the y-direction while x is periodic. The box is taken as $[\lambda \times 3\lambda]$ to limit boundary effects with wallslip imposed at the top and bottom as shown in Fig. 6.2.
The error is defined as the RMS of relative amplitude error over time

$$L_2(A) = \sqrt{\frac{1}{T\omega_0}\int_0^T \frac{|(A_{exact}(\tau) - A(\tau)|}{A_0} d\tau} \quad , \tag{6.40}$$

with $T\omega_0 = 25$ which corresponds to approximately 4 oscillations.
The error $L_2(A)$ is displayed in Fig. 6.17d with a number of points in the wavelength N_λ from 8 to 128. All methods converge at second-order with a better accuracy for VOF. A convergence saturation is observed for SLS which is due to parasitic perturbations appearing for the highest resolution. A temporal evolution of the amplitude for $N_\lambda = 32$ is also presented in Fig. 6.17c, where VOF is already very accurate while SLS, CLSVOF and ACLS are a little bit shifted. Finally, the CLSVOF method exhibits higher oscillation



(a) $N_\lambda = 8$ **(b)** $N_\lambda = 16$

(c) $N_\lambda = 32$ **(d)** $L_2(A)$ *error*

Figure 6.17: *Temporal evolution of* A/A_0 *for different mesh resolutions and mesh error convergence of* $L_2(A)$

amplitudes for the coarse meshes. To better understand this behaviour, another simulation has been performed using HF with CLSVOF for curvature computation leading to results equivalent to VOF results. We conclude that the higher amplitude is due to the curvature computation. One explanation would be the initial perturbation of ϕ through the coupling between LS and VOF : f and ϕ are not exactly matching at the initialization and the first iterations introduce an amplification of the oscillation due to the coupling.
Note that this test case is specifically well suited for VOF as the HF method is the most accurate in mesh-aligned configurations. This explains why it performs better than the other methods on the damping wave.

6.2.2 Momentum conservation

Now that interface capturing methods have been compared on imposed velocity fields, the coupling with the two-phase solver is explored. To assess the momentum conservation of our solver, the classic density ball test case introduced in [25] is presented where a 2D droplet

of radius 0.1 and density 10^6 is translating at a velocity $u = 1$ m/s in a unity density field at rest. The density is high enough to consider the transport as a pure solid translation. The error E_{shape} of Section 4.2.1 is used here to quantify the interface transport. A new error E_{tke} is introduced to evaluate momentum conservation with the same formalism as E_{mass}

$$E_{tke} = \frac{1}{K_0 T} \int_0^T |\Delta K| dt \quad , \tag{6.41}$$

with $\Delta K = K(t+dt) - K(t)$ the variation of kinetic energy computed at time t and $t+dt$. With K computed as in [238]

$$K = \frac{1}{2} \sum_{i=1}^{N_\mathcal{C}} \rho_i \|\mathbf{u}_i\|^2 \mathcal{V}_i \quad , \tag{6.42}$$

with $\rho_i = \rho_g + H^\epsilon_{\Gamma,i} [\rho]_\Gamma$ and $\mathbf{u}_i$ components defined as average of face velocities.

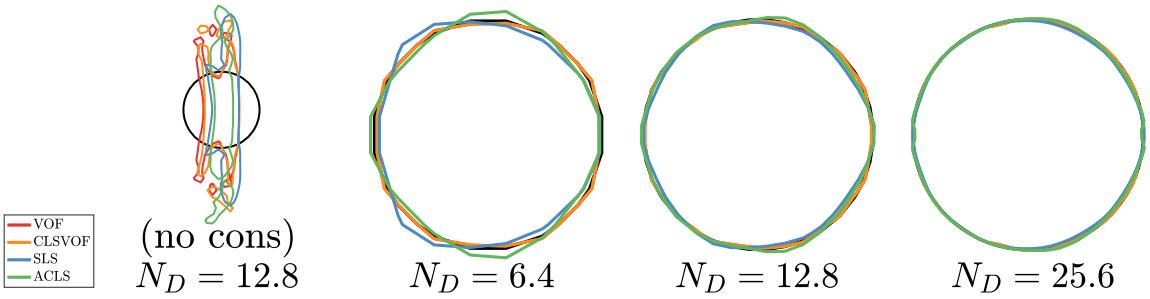
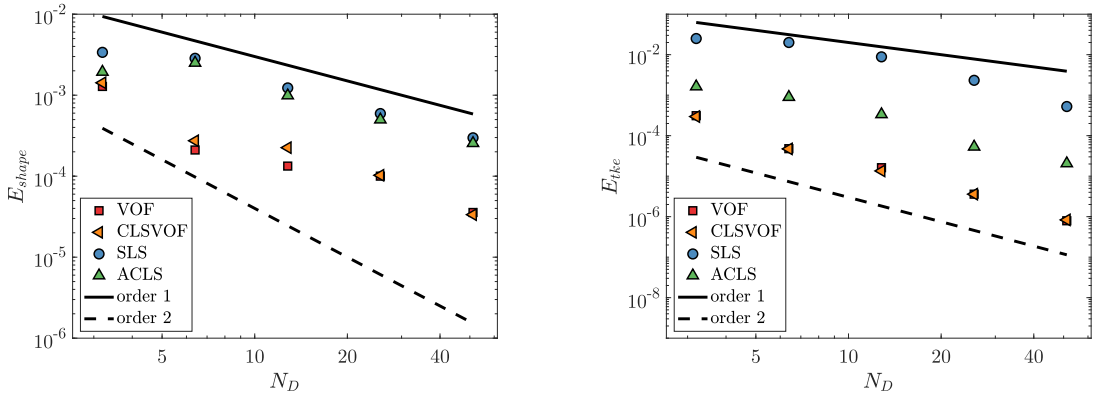


Figure 6.18: *Initial and final shape for the density ball translation, the first image at the left shows a case without the consistent scheme while the other images are zoomed on the circle shape*

Fig. 6.18 displays the shape for different meshes. A non-conservative form of the momentum transport has been added for completeness where the shape is not conserved at all. For all other simulations, using the momentum fix, the initial circular shape is well preserved by the VOF and CLSVOF approaches even for very low resolution. This can be explained by the density approximation based on PLIC which is more accurate and less diffusive than Eq. (6.12). However, the circle is less distorted than the one presented in [47] where Eq. (6.11) was used coupled with a ACLS approach.
When looking at more quantitative metrics, all methods display a convergence rate between 1 and 2 for E_{shape} in Fig. 6.19a with a better accuracy for VOF and CLSVOF. This shape error is explained by the difference in momentum conservation. A huge difference of two order of magnitude is observed between VOF and SLS for E_{tke} in Fig. 6.19b. It is also interesting to notice that ACLS is more conservative than SLS for momentum too. This illustrates how momentum conservation is impacted by the choice of ρ_u^n computation and mass conservation.

6.3 Applications



(a) E_{shape} *error* **(b)** E_{tke} *error*

Figure 6.19: *Mesh error convergence for the density ball translation*

6.3.1 Droplet collision

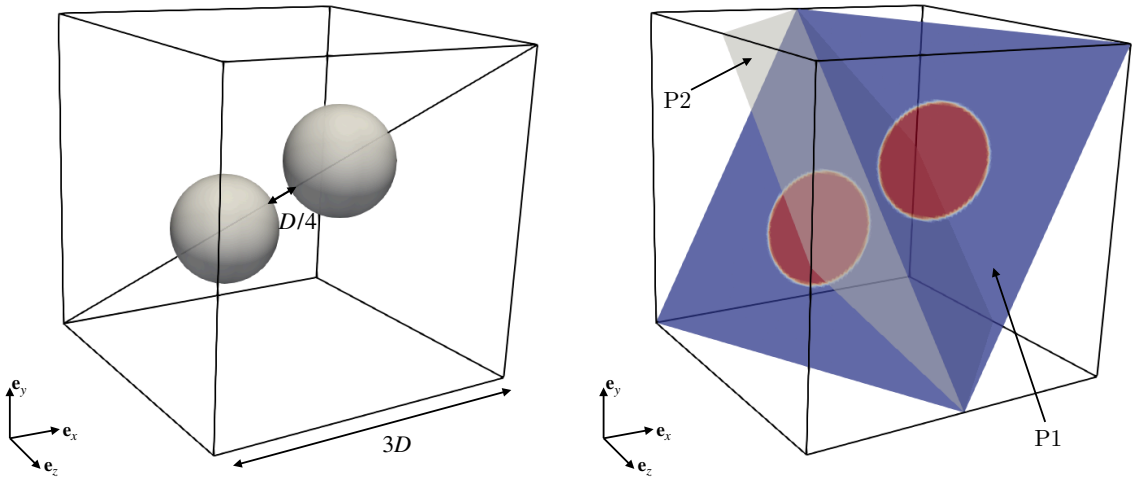
A collision between two water droplets in quiescent air is presented. The solver have to handle large density ratio and strong capillary effects accurately in order to retrieve the correct collision regime. Indeed, a bad prediction of curvature or momentum will lead to a modification of the Weber number $We = \frac{\rho D u^2}{\sigma}$ and the Ohnesorge number $Oh = \frac{\mu}{\rho D \sigma}$ which drive the collision regime. Here, the head-on collision of two equal-sized droplet with $We = 40$ and $Oh = 0.0047$ is considered. The expected outcome is a reflexive collision with one satellite observed experimentally [8]. This regime includes coalescence and break up which are the most challenging behaviour to capture as they will always happen in the mesh resolution limit. Thus, this last test case is discriminating for interface capturing methods as they behave differently in this limit cases. Special attention is drawn to the topology transitions.

The set up is the same as in [61]: the two droplets are located in the body diagonal of of $[3D \times 3D \times 3D]$ cube with a distance of $D/4$ between each other as illustrated on Fig. 6.20a. A resolution of $N_D = 40$ is chosen and the boundary conditions are free-slip walls in all directions. This diagonal droplet trajectory has two main interests: it avoids any favourable alignments with the mesh and it allows to take a smaller domain to reduce computational time.

In Fig. 6.21 are displayed the four topology changes :

- Coalescence of the droplets at $t_1 = 150$ μs
- Film break up of the disk at $t_2 = 1250$ μs
- Torus coalescence at $t_3 = 1850$ μs
- Thin cylinder break up at $t_4 = 4700$ μs

Each topology change is characterized by a mass change for SLS and ACLS and a peak of $\|\mathbf{u}\|_{max}$ for all methods (see Fig. 6.22). This is because of the bad curvature and normal computations when the mesh limit is reached, which is always the case in topology changes. The expected satellite is retrieved at the end of the simulation even if the size vary depending



(a) *Initial setup* **(b)** *Plane definition*

Figure 6.20: *Numerical configuration for the head-on collision*

on the method used.
The first coalescence at t_1 introduces small bubbles at the stagnation point for VOF and CLSVOF because of the normal computation. As the mass is conserved with VOF and CLSVOF, these bubbles will be trapped in the liquid (see Fig. 6.23). It is also interesting to notice that ACLS also creates bubbles as previously observed in [90]. However, they represent less gas mass compared to VOF and CLSVOF. The presence of these bubbles can cause numerical problem in the simulation as it is not well-resolved by the mesh and produces very bad curvature evaluations. This effect can be related to the spurious $\|\mathbf{u}\|_{max}$ behaviour for VOF in Fig. 6.22b and the use of CLSVOF seems to reduce this phenomenon. In the case of SLS, these bubbles are rapidly turning to liquid as they go under mesh resolution, this corresponds to the first mass creation at t_1 in Fig. 6.22a.
At t_2, the film is so thin that it goes under mesh resolution which creates break up. This break up causes a loss of mass for ACLS while the SLS manage to maintain the film for a longer time by creating even more liquid mass. VOF, CLSVOF and ACLS create small under-resolved structures at the mesh limit when the film is too thin. This behaviour is inherent to the methods and is the same as for the sphere deformation case of Sec. 4.3.3.
The torus coalescence at t_3 implies the highest peak in $\|\mathbf{u}\|_{max}$ as a lot of small structures created during the film break up are merging and creating multiple source of momentum trough curvature contribution.
Finally all this event history of curvature computation failure and mass change has an impact on the final outcome observed at t_4. The satellite is of different size and for VOF, not centred any more.
In Fig.6.24 are compared the satellite break up obtained with the different methods and an experimental acquisition of [8]. The satellite mass is under-predicted while the two droplet

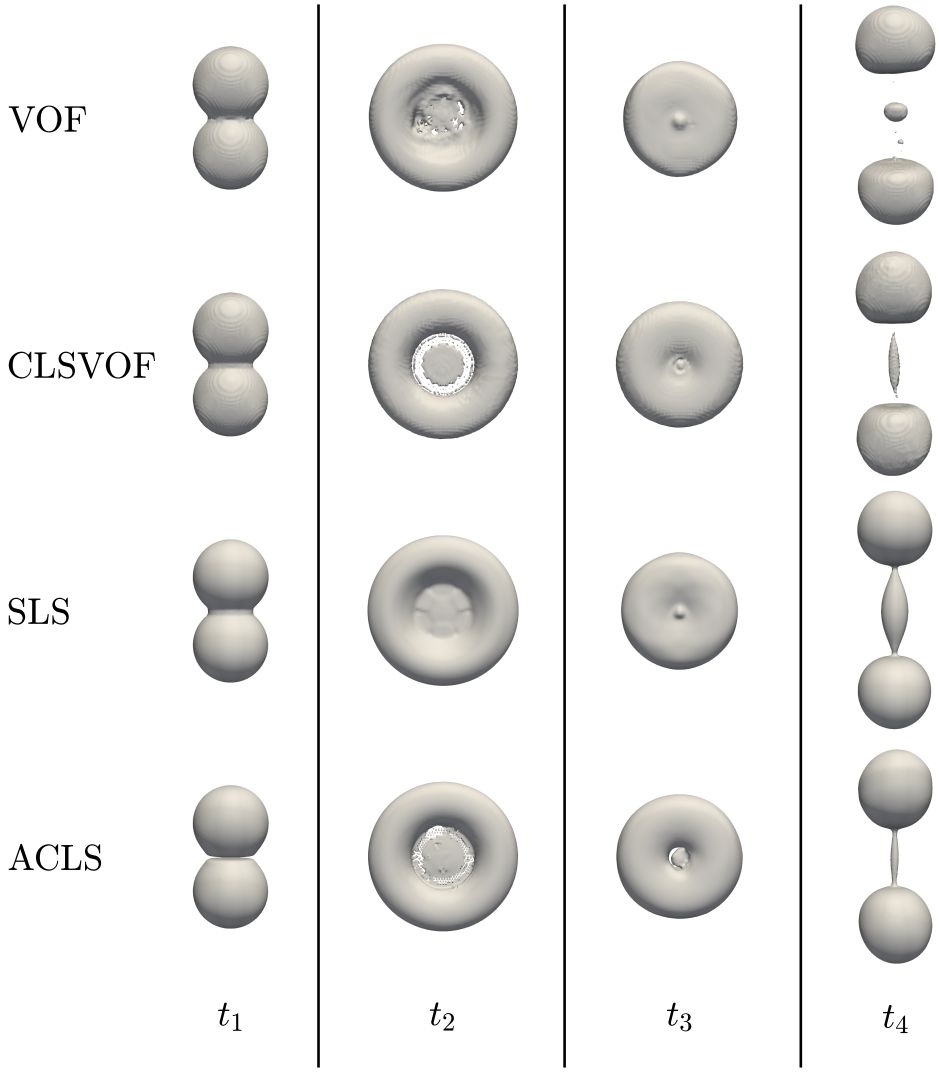
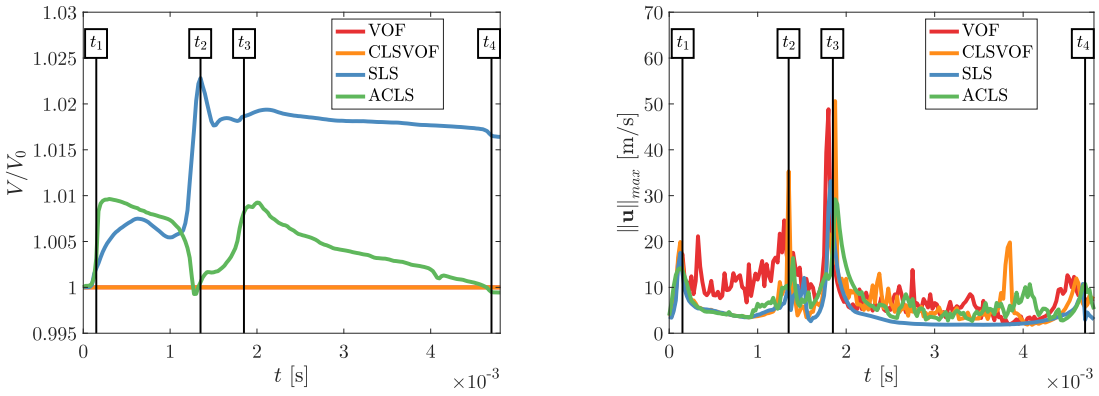


Figure 6.21: *Shape for the four topology changes during the head on collision, t_1 and t_4 are represented on P1 while t_2 and t_3 are on P2 (see Fig. 6.20b for plane definition)*

mass is over-predicted as observed in [217] using SLS. This can be caused by capillary instabilities causing premature break-up compared to experiment. The SLS seems produces the biggest satellite but this is partly due to the mass gain of around 1.5 % observed in Fig.6.22a. The CLSVOF seems to provide a good trade off between conservation and accuracy.

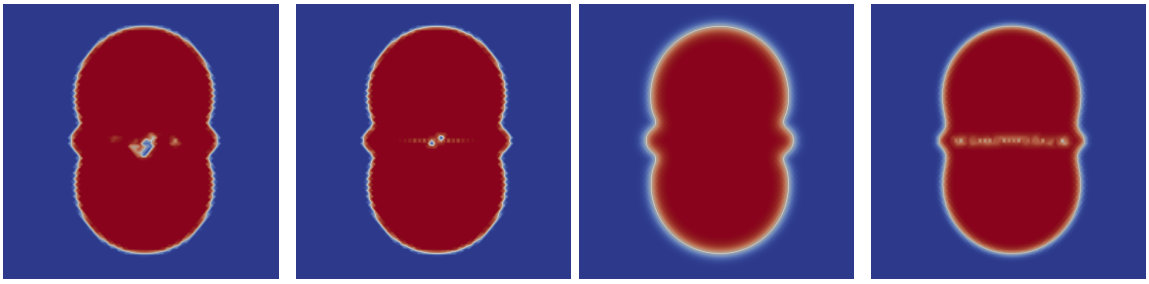
6.3.2 Shear layer

This last test case is a planar shear flow similar to [10] and reproduces critical aspects of complex atomization configurations. Without regularizing effect such as viscosity or surface



(a) V/V_0 **(b)** $\|\mathbf{u}\|_{max}$

Figure 6.22: *Temporal evolution of* V/V_0 *and* $\|\mathbf{u}\|_{max}$ *during the head on collision*



(a) *VOF* **(b)** *CLSVOF* **(c)** *SLS* **(d)** *ACLS*

Figure 6.23: *Collision outcome at* $t = 2500$ μs *represented here with* H_Γ^ϵ *on P1*

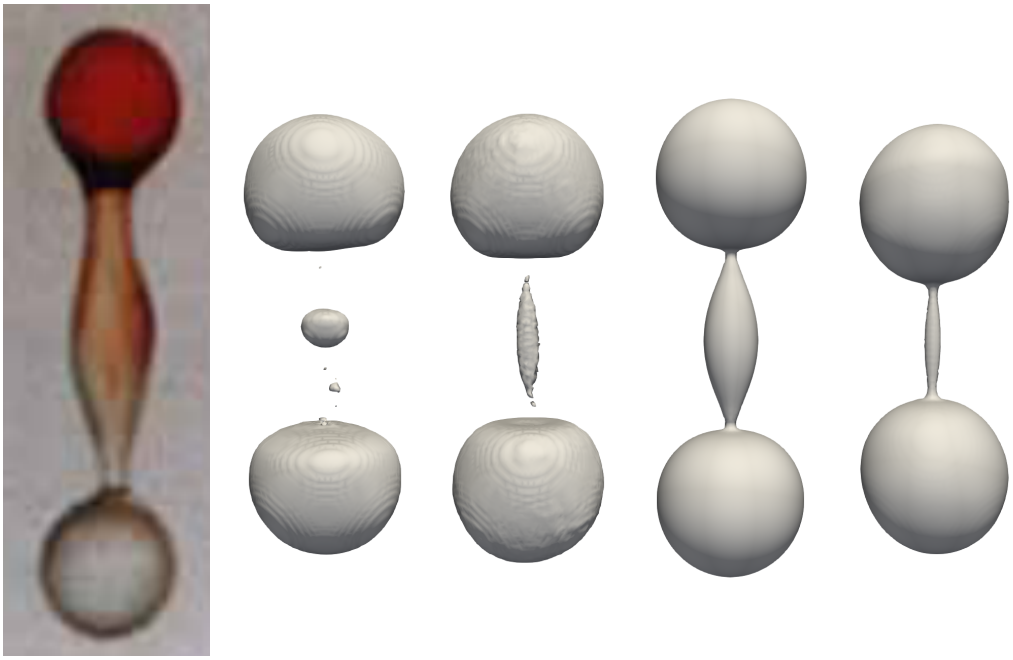
tension ($We = \infty$ and $Oh = 0$), the jet will break up in thin structures only impacted by convection. Small errors in the computation of ρ_f lead to error in velocity which can cause severe stability issues. In [64], some of the methods could not reach a long physical time before the simulation breaks down. The set up is illustrated in Fig. 6.25 with the periodic box length $L = 1$ mm and the liquid shear layer thickness $\delta = L/10$. Here, the densities are chosen such that $\rho_l/\rho_g = 1000$ and the initial divergence-free velocity is defined as

$$u = U_0 - 0.04\frac{L}{2\pi}\frac{-4y}{\delta^2}\cos\left(\frac{2\pi x}{L}\right)\exp\left(-2\left(\frac{y-h_0}{\delta}\right)^2\right)\text{ m/s} \quad , \tag{6.43}$$

$$v = 0.04\sin\left(\frac{2\pi x}{L}\right)\exp\left(-2\left(\frac{y-h_0}{\delta}\right)^2\right)\text{ m/s} \quad , \tag{6.44}$$

with U_0 is 2 m/s in the liquid and 30 m/s in the gas and $h_0 = L/5$ the liquid shear center position.

The simulation is performed until one of the methods fails. In our framework, all methods seemed to be robust but the ACLS which breaks down after $t = 1.65$ ms, while the others manage to reach the same physical time $t = 2$ ms as in [10]



(a) Ashgriz et al. (b) VOF (c) CLSVOF (d) SLS (e) ACLS

Figure 6.24: *Satellite break up comparison with experiment*

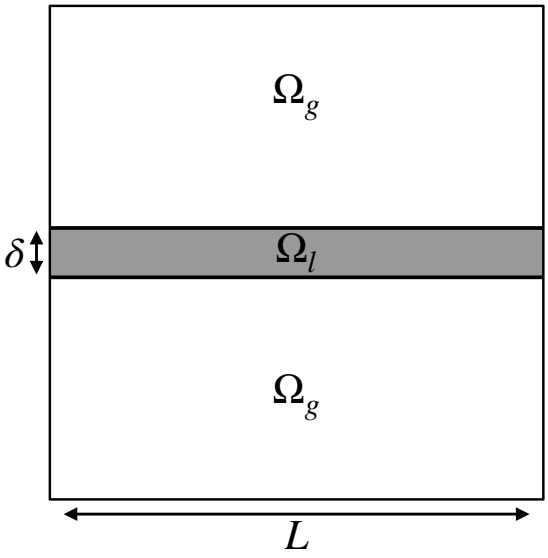
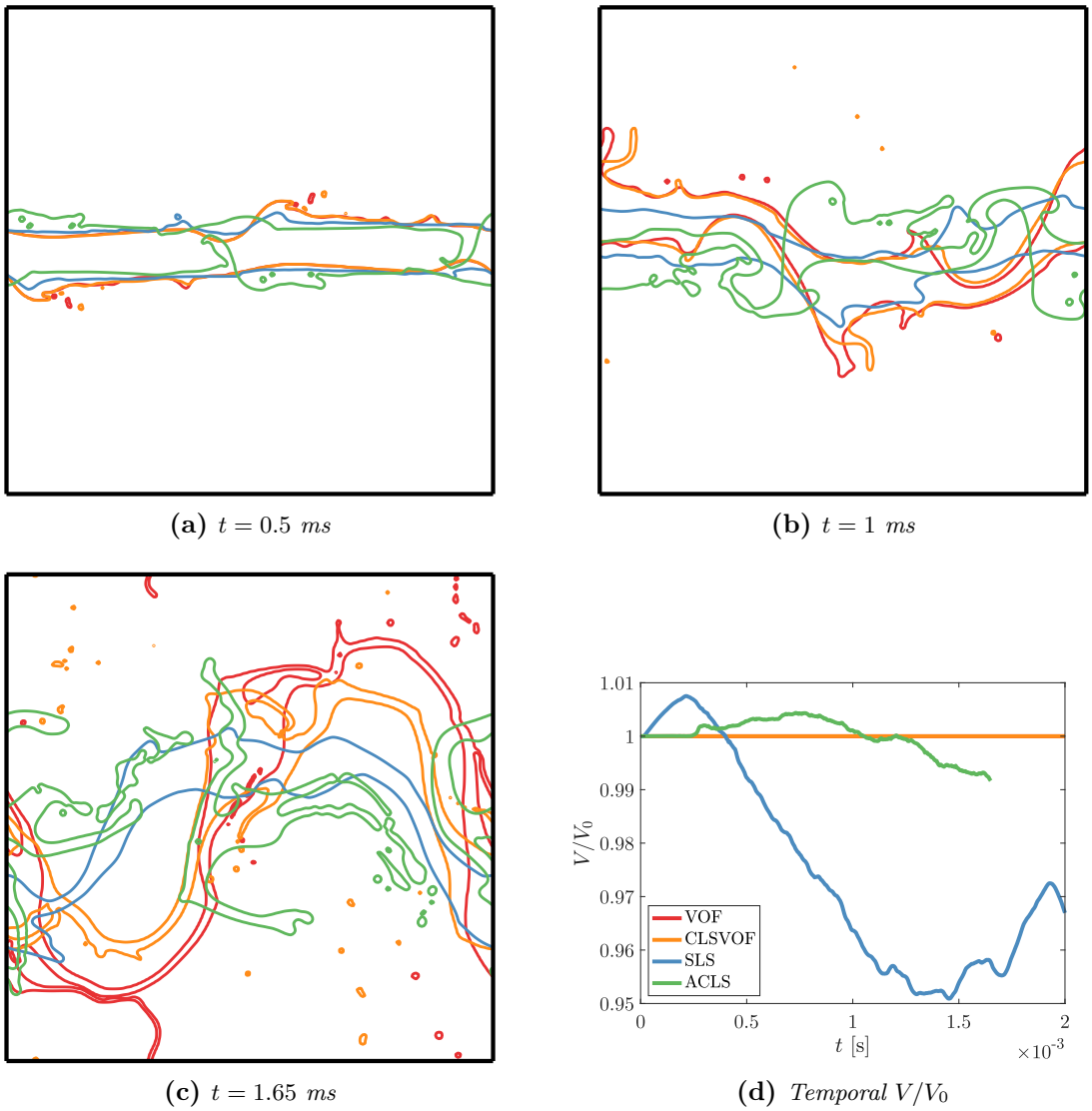


Figure 6.25: *Shear layer simulation set up*

In Fig. 6.26 are given isocontour of the four interface capturing methods at different times. While VOF and CLSVOF are behaving similarly at the beginning, the discrepancy is in-



(a) $t = 0.5$ *ms* **(b)** $t = 1$ *ms*

(c) $t = 1.65$ *ms* **(d)** *Temporal* V/V_0

Figure 6.26: *Liquid layer isocontour for* ***(a)*** $t = 0.5$ *ms,* ***(b)*** $t = 1$ *ms,* ***(c)*** $t = 1.65$ *ms for VOF (red line), CLSVOF (orange line), SLS (blue line) and ACLS (green line) and* ***(d)*** *the temporal evolution of mass for the shear layer test case*

creasing with time. Moreover, SLS and ACLS are showing even more different behaviour compared to VOF-based methods. In fact, SLS does not create under-resolved structured while droplets of the mesh size are present for all other methods. This shows how interface capturing methods impact the topology of a liquid jet under high convective effects when the structure are not properly captured anymore. In this test case it is difficult to conclude on what method is the best, even if the ACLS exhibits some stability problems. This issue arises when very thin structures are created and density is approximated poorly, leading

to very high velocities without any diffusive process to damp them. It is also important to notice that SLS while robust has lost about 4% of mass during the whole simulation time while ACLS managed to maintain a mass error above 1%. This last test case enlightens that under-resolved structures behaviour which are present in atomization simulations will be highly impacted by the interface capturing choice and totally driven by the mesh size. The outcome at a large physical time is thus completely different.

6.4 Conclusion

The comparison of four popular methods of the literature has been presented focusing on mass and momentum conservations, and geometrical accuracy. Overall, the interface capturing methods presented here are able to provide a good physical description of two-phase flows with high-density ratio and capillarity effects, with specific strengths and weaknesses. While VOF and CLSVOF are exactly mass conservative and show good momentum conservation, SLS provides a representation of the interface able to compute normal and curvature easily and with more robustness even in dynamic cases thanks to the smoother representation of the interface. ACLS shows improvements in the mass and momentum conservation compared to SLS at the cost of a loss of geometrical accuracy which can be very severe at high resolution. For coarse to medium resolutions, ACLS accuracy is slightly better than VOF or CLSVOF for dynamic cases. The head-on collision has demonstrated the robustness of all methods, showing that they are all able to retrieve the satellite droplet. However, the atomization case has also shown that the mesh resolution has a great impact on the accuracy and the development of thin structures, all methods giving very different results for long times of simulation. This calls for further in-depth statistical analysis in such chaotic test cases.
Overall, Coupling VOF with LS seems to be the most promising choice in a cartesian finite-volume framework as the conservation properties of VOF are preserved while curvature and normal are easier to compute and more robust. The versatility of such strategy, will be particularly of interest when additional physics is added, such as phase change. Our unified framework is in that sense the adequate vessel for such developments.
It has to be reminded that our conclusions hold for this specific unified framework. Indeed, the under-resolved problems met in the head-on collision could be better handled by using local mesh refinement [169; 81; 90]. Moreover, our VOF and CLSVOF transport rely on dimensional splitting and the geometrical operations are straightforward on a Cartesian mesh. This is no longer true in unstructured meshes, and the ACLS method can be a good alternative to keep good accuracy and conservation.
The next part investigate the inclusion of additional physics in the solver to take into account the phase change. The aim is to verify if the conclusions still hold for an incompressible two-phase flow solver with phase change.

Part IV

Simulation of two-phase flows with phase change

Chapter 7

Numerical challenges of simulations with phase change

Contents

From now, the manuscript has presented the methodology to build an incompressible two-phase flow solver without phase change. In such case, the main difficulty arisen from the curvature computation and the momentum conservation. However, the final objective of the work is to study evaporating droplets, requiring to include additional physics in the solver, and so, additional numerical difficulties. The occurrence of phase change in a flow leads to important modifications in the governing equations with the presence of new discontinuities at the interface, as presented in Chapter 1. It is also important to keep in ind that some of the equivalence between WDF and JCF presented in Chapter 5 are no longer true, because

of the difference in the averaging process discussed in Section 2.6.
The main challenges emerging from the integration of phase change in an incompressible two-phase flow solver are the following:

1. The computation of the evaporation rate $\dot{m}$, which is the pillar of the phase-change phenomenon: as for the evaluation of κ to obtain an appropriate surface tension modelling, the evaluation of $\dot{m}$ requires specific attention to properly describe phase change. Moreover, it appears in numerous terms such as velocity jump, interface regression or source term of energy or species mass which imply a very tight coupling between the equations. Note that reconstruction of other interface quantities such as the interface temperature T_Γ or the species mass fraction at the interface Y_Γ could also be required in some numerical approaches and need to be evaluated accurately.
2. The treatment of the velocity jump $[\mathbf{u}]_\Gamma$ in all transport equations:
   (i) The continuity equation cannot be reduced to a simple divergence-free condition $\nabla \cdot \mathbf{u} = 0$. As presented in Chapter 2, the jump in velocity appears in the divergence of velocity located at the interface which needs to be numerically prescribed in the projection method.
   (ii) The momentum equation needs to be solved accordingly to include the jump in velocity. This can require special treatments of the convective and diffusive operators depending on the formulation used.
   (iii) The transport of the interface is no longer performed with the velocity of the flow as it differs from the velocity of the interface. In fact, the velocity of the interface needs a reconstruction to include the regression caused by phase change.
   (iv) Scalar equations need to treat the convective operator to handle the velocity discontinuity at the interface.
3. The temperature T needs to be solved as it is used to close the system of equations. The main difficulty arises in the heat flux jump at the interface due to phase change which needs to be taken care of.
4. The species mass fraction Y also needs to be solved to close the system of equations. In the case of pure liquid, it is only defined in the gas and needs proper definition of the source of species mass at the interface.

The aim of this chapter is to give details on these challenges with a state-of-the-art of the simulation of incompressible two-phase flows with phase change. The works are systematically related to the formulations derived in Chapter 2 to clarify the solver choices in the literature.
In Section 7.1, general considerations on the possible reconstructions of phase-change quantities at the interface are presented. Then, the velocity discontinuity handling is described in Section 7.2 in both prediction step (momentum equation) and projection step (continuity equation). Section 7.3 gives details about the integration of interface regression in the interface capturing methods. Finally, Section 7.4 presents a state-of-the-art on the resolution of energy and species mass fraction equations with special attention on the formulation used to take into account the interface contributions.

7.1 Reconstruction of phase-change quantities

The phase-change procedure needs a closure for the quantities at the interface which are the evaporation rate $\dot{m}$, the interface temperature T_Γ and the species mass fraction at the interface Y_Γ. While $\dot{m}$ appears in all jump conditions related to phase change, the interface temperature and species mass fraction can be used explicitly in a phase-change solver at different stages to:

- Impose boundary conditions or jump conditions at the interface;
- Compute accurate gradients in the interface neighbourhood for the computation of $\dot{m}$.

A general system of equation can be written to determine these quantities by using the jump conditions and the thermodynamic relation between T_Γ and Y_Γ

$$\begin{cases} \dot{m} = \dfrac{[k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma}{[h]_\Gamma} \quad , & \text{(7.1a)} \\ \dot{m} = \dfrac{\rho_g \mathcal{D}_v \left.\nabla Y\right|_\Gamma \mathbf{n}_\Gamma}{Y_\Gamma - 1} \quad , & \text{(7.1b)} \\ Y_\Gamma = \dfrac{P_s\left(T_\Gamma\right) M_v}{P_s\left(T_\Gamma\right) M_v + \left(P_{ref} - P_s\left(T_\Gamma\right)\right) M_0} \quad . & \text{(7.1c)} \end{cases}$$

If $[k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma$ and $\left.\nabla Y\right|_\Gamma$ include explicitly the values of T_Γ and Y_Γ at the interface, then the system is closed.
In the boiling regime, the gas only consists of the pure vapour. Then, $Y_\Gamma = 1$ and T_Γ is imposed at the pure liquid saturation temperature T_{sat}. Hence, the system Eq. (7.1) reduces to

$$\begin{cases} \dot{m} = \dfrac{[k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma}{[h]_\Gamma} \quad , & \text{(7.2a)} \\ T_\Gamma = T_{sat} \quad . & \text{(7.2b)} \end{cases}$$

For general vaporization problems, the system Eq. (7.1) needs to be solved. For clarity, the relations for $\dot{m}$ Eqs. (7.1a) and (7.1b) are defined as $\mathcal{M}_T(T_\Gamma)$ and $\mathcal{M}_Y(Y_\Gamma)$ respectively. The Clausius-Clayperon relation Eq. (7.1c) is referred as $\mathcal{R}(T_\Gamma)$ and $\mathcal{R}^{-1}(Y_\Gamma)$ for the reverse relation.
The system can be reformulated into an implicit equation on T_Γ

$$T_\Gamma = \mathcal{R}\left(\mathcal{M}_Y^{-1}\left(\mathcal{M}_T(T_\Gamma)\right)\right) \quad . \tag{7.3}$$

Numerically, $\mathcal{M}_T$ and $\mathcal{M}_Y$ contain also the normal gradients $T_l^{(1)} = \nabla T_l \cdot \mathbf{n}_\Gamma$, $T_g^{(1)} = \nabla T_g \cdot \mathbf{n}_\Gamma$ and $Y^{(1)} = \nabla Y \cdot \mathbf{n}_\Gamma$ which are not known a priori and needs an explicit reconstruction with T_Γ and Y_Γ appearing in the stencil. This can be done by simple differencing as in [205; 159] or by fitting an Erf function as in [122]. However, Palmore et al. [159] pointed out the necessity to have accurate treatment of $\dot{m}$ to obtain converging methods. This is why, in most of the methodologies presented in the literature, $\dot{m}$ is obtained directly from $\mathcal{M}_T$ or $\mathcal{M}_Y$ by using more sophisticated normal gradient evaluations.

7.1.1 Evaporation rate closure

Letting aside the general phase-change system Eq. (7.1), one of the expressions of $\dot{m}$ has to be used arbitrarily. While this approach leads to the loss of the discrete equality $\mathcal{M}_T = \mathcal{M}_Y$ it gives more degree of freedom on the computation of the normal gradients required in the expressions of $\dot{m}$. More specifically, it does not have to contain the interface value anymore. The reconstruction of accurate gradients at the interface is a complex numerical problematic which will be detailed in Section 8.2.

7.1.1.1 Species mass fraction gradient based

The approach used in most of the evaporation solvers [218; 195; 205; 122; 192] is based on $\mathcal{M}_Y$. as it only requires one gradient computation and the knowledge of Y_Γ at the interface.
While it gives a straightforward framework for the simulation of general unsteady evaporation process, it suffers from ill-posedness when the interface conditions are close to $T_\Gamma = T_{sat}$ and $Y_\Gamma = 1$. This has been first observed in [205] which motivated the resolution of Eq. (7.1) to compute $\dot{m}$ instead. They noticed that this problem arises in the case of high evaporation rate only. This behaviour has been widely explored in [29] where in the case of high evaporation rate, the use of $\mathcal{M}_Y$ to compute $\dot{m}$ is prompt to large errors when $Y_\Gamma > 0.9$. This is explained by the amplification of numerical errors on $Y_v^{(1)}$ as $1 - Y_\Gamma$ is closer to zero. The error can be reduced by decreasing the timestep. However this is not an acceptable solution for HPC evaporation computations and other approaches need to be used to address this issue.

7.1.1.2 Temperature gradient based

A way around is to use the other expression of $\dot{m}$ defined by $\mathcal{M}_T$. This is the starting point of all the boiling solvers [204; 33; 219; 7; 236; 123; 187] and it can be generalized to evaporation simulations as in [190; 183; 159] by considering an interface temperature $T_\Gamma \neq T_{sat}$. This allows to have a formulation which can handle both evaporation and boiling without further considerations.
The only limitation of this approach is the accuracy of phase gradients evaluated at the interface. When a WDF is used for the temperature, the temperature values in Ω_Γ are enthalpy-averaged as demonstrated in Chapter 2. In the vicinity of the interface, it is then not straightforward to properly define both $T_l^{(1)}$ and $T_g^{(1)}$ at the same location from an averaged information of temperature. This will be further detailed in the gradient computation study presented in Section 8.2 and the two-fluid approach of Section 7.4.3.

7.1.1.3 Regime switch

Finally, in the work presented in [29], a switch between the two regimes is proposed to benefit from both formulations and avoid the numerical issues previously encountered. If T_Γ is below a certain value T_c, the interface is supposed to be far enough from boiling and $\mathcal{M}_Y$ is used for $\dot{m}$ computation. When T_c is reached, the method switches to the boiling

limit where $\dot{m}$ is computed from $\mathcal{M}_T$ instead. From numerical investigations, T_c is defined such that $Y_\Gamma = 0.9$, in other words, $T_c = \mathcal{R}^{-1}(0.9)$.
Other authors also considered the use of both formulations in the same solver [198; 85]. But no switch is possible during a given simulation, the formulation is chosen with respect to the test case simulated.

7.1.2 Temperature and species mass fraction at the interface

As discussed previously, T_Γ and Y_Γ can be retrieved from solving the general system Eq. (7.1). However, if $\dot{m}$ is evaluated from a methodology of Section 7.1.1, the relevance of solving iteratively Eq. (7.1) is limited as the system will not be verified ($\mathcal{M}_T \neq \mathcal{M}_Y$). This is why most of the phase-change solvers use a reduced system of equation to obtain T_Γ and Y_Γ [195; 183; 218; 29; 192]. Therefore, the only constraint imposed on the interface quantities is the relation Clausius-Clayperon relation $Y_\Gamma = \mathcal{R}(T_\Gamma)$ and another equation is required to close the system. This is done by either deducing T_Γ from the computed temperature field T or Y_Γ from the computed species mass fraction field Y through an extrapolation operator $\mathcal{E}$. The two possible systems of equations are then

$$\begin{cases} Y_\Gamma = \mathcal{R}(T_\Gamma) \\ T_\Gamma = \mathcal{E}(T) \end{cases}, \tag{7.4}$$

$$\begin{cases} T_\Gamma = \mathcal{R}^{-1}(Y_\Gamma) \\ Y_\Gamma = \mathcal{E}(Y) \end{cases}. \tag{7.5}$$

In [195], the whole computational cell is considered at saturation and the interface temperature is directly taken as $T_\Gamma = T^H$ with T^H the WDF temperature. A similar approximation is used in [183] to obtain Y_Γ from Y. It results in the first-order extrapolation $\mathcal{E}_1$. In [218; 29; 192], the liquid temperature is extrapolated linearly using the subcell distance to the interface $\theta\Delta x$ to retrieve T_Γ resulting in the second-order extrapolation $\mathcal{E}_2$. Note that basing the extrapolation on the liquid temperature is an ad-hoc choice and gas temperature could be used instead. More details are given on the discretization of these extrapolations in Section 8.2.

7.1.3 Conclusion

From the above considerations, different methodologies seem to be relevant for building an accurate phase change procedure. The critical aspect is the reconstruction of the gradients at the interface required for computing $\dot{m}$. The choice of keeping the general system Eq. (7.1) seems to be the most relevant from a physical perspective. However, it implies more numerical constraints on the gradient reconstruction (the interface quantity has to be in the stencil to close the system). Then, the accuracy of $\dot{m}$ can be limited leading to inconsistent phase-change procedures. All these aspects are treated in Section 8.2 with a complete study of the gradient reconstruction which allows to conclude on the best-suited methodology for the reconstruction of phase change quantities.
On the other hand, letting aside the general system alleviates the constraint on the gradient reconstruction, allowing more accurate gradient reconstructions. However, the

reduced system requires a numerical closure based on the scalar fields which leads to a strong coupling of the interface quantities to either T or Y. It will be seen in Section 9.2.1.3 that this can lead to limitations in the numerical behaviour for phase change simulations.

7.2 Velocity discontinuity

Because of the phase change occurring at the interface, the velocity is no longer continuous and the projection method procedure has to be adapted accordingly. First, the jump in velocity needs to be prescribed for the updated velocity $\mathbf{u}^{n+1}$ through the pressure correction Eq. (5.3). Then, the discontinuity of the velocity also appears in the momentum equation and have to be treated in the prediction step Eq. (5.1). As a reminder, the jump in velocity is expressed as

$$[\mathbf{u} \cdot \mathbf{n}_\Gamma]_\Gamma = \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \tag{7.6}$$

7.2.1 Velocity jump in the continuity equation

When velocity is discontinuous across the interface, a treatment is required to take this into account in the continuity equation. This equation is not explicitly advanced in the projection method, however, it appears indirectly in the Poisson equation for pressure. Rewriting Eq. (5.3) without the simplification $\nabla \cdot \mathbf{u}^{n+1} = 0$ leads to

$$\nabla \cdot \left(\frac{1}{\rho} \nabla P\right) = \frac{1}{\Delta t} \left(\nabla \cdot \mathbf{u}^* - \nabla \cdot \mathbf{u}^{n+1}\right) \quad . \tag{7.7}$$

The goal is then to evaluate $\nabla \cdot \mathbf{u}^{n+1}$ accounting for the velocity jump.

7.2.1.1 Whole domain formulation

The WDF for the continuity equation leads to a source term to model the velocity jump. This formulation is used in several work with Front Tracking [58; 85; 116], Level Set [204; 121; 219] or VOF [227; 190; 236; 192; 159; 52]. This results in the following expression of the divergence operator applied on $\mathbf{u}^{n+1}$

$$\nabla \cdot \mathbf{u}^{n+1} = \dot{s}_\rho = -\dot{m} \left[\frac{1}{\rho}\right]_\Gamma \delta_\Gamma \quad . \tag{7.8}$$

Then the Dirac definition can take several forms depending on the authors. We recall that the notation is notional when it comes to VOF, as the operator ∇ has to be integrated over a cell volume.

Simple smooth approach Some VOF approaches [192; 52] rely on the definition

$$\delta_\Gamma = |\nabla f| \quad , \tag{7.9}$$

which is the analogous CSF treatment for velocity jump.
This method is straightforward to implement but tends to smear the velocity jump on several cells and result in large errors in the velocity field [219; 107].

Sharp approach To produce more accurate results for the velocity field, other works base their Dirac computation on geometric arguments [190; 236]. Based on the surface area, the Dirac is then defined as

$$\delta_\Gamma = \frac{A_\Gamma}{V_\mathcal{C}} \quad . \tag{7.10}$$

This expression is homogeneous to $[L]^{-1}$ and represent a very sharp source term only located at the interface.
Another sharp approach was more recently proposed in [159]

$$\delta_\Gamma = \frac{Q}{\Delta x} \quad , \tag{7.11}$$

where Q is one only in cells containing the interface. This expression is a first-order representation of the Dirac which considers a constant interface area in all mixed cells. The sharp Dirac distribution leads to accuracy improvement for the velocity field. However, this formulation acts as a point source term in Eq. (7.7) and leads to a very stiff problem which can suffer from instabilities [236]. In [107], large oscillations of pressure are observed for the simulation of condensing bubbles by using similar formulations in a Level Set framework.

Smoothing of the sharp Dirac In contrast to the sharp formulations described before, Hardt and Wondra [76] propose to smooth $\dot{s}_\rho$ over few cells by solving a static diffusion equation of the form

$$\nabla \cdot \left(\epsilon \nabla \widetilde{\dot{s}}_\rho \right) = \dot{s}_\rho \quad , \tag{7.12}$$

with ϵ a controlled smearing width.
It has also been done in [236] with Eq. (7.10) to define the sharp source term

$$\dot{s}_\rho = -\dot{m} \left[\frac{1}{\rho} \right]_\Gamma \frac{A_\Gamma}{V_\mathcal{C}} \quad . \tag{7.13}$$

Note that in this procedure, the whole source term is diffused instead of the Dirac only, which creates also a diffusion of the term $\dot{m}$. By using such approach, the velocity jump smearing is controlled and the total source of mass is conserved such that $\int_\Omega \dot{s}_\rho d\Omega = \int_\Omega \widetilde{\dot{s}}_\rho d\Omega$. Numerical experiments in [236] showed cancellation of large pressure oscillations without loss of accuracy.

Mass-average correction For the problem of spurious pressure oscillations, another explanation is provided based on the mathematical formulation of $\dot{s}_\rho$ arising from the WDF of governing equations instead of the numerical considerations given in [236]. In [195], an important point is raised about the use of the volume source $\dot{s}_\rho$ applied to the divergence of a velocity which is mass averaged in the WDF as derived in Chapter 2. An iterative algorithm is thus employed to apply a correction on the source term $\dot{s}_\rho$ to take into account the mass averaged nature of the velocity. The use of this procedure showed huge reduction in the pressure oscillations and a less spurious behaviour of the Sherwood number in the case of evaporating droplet simulations. It has also been used in the case of boiling bubbles [74].
Later, a more straightforward approach has been proposed in [122] by using simple definitions. First, $\mathbf{u}^M$ and $\mathbf{u}^V$ are related through

$$\mathbf{u}^M = \mathbf{u}^V + [\rho]_\Gamma \frac{f(1-f)}{\rho^V} [\mathbf{u}]_\Gamma \quad , \tag{7.14}$$

then the correct divergence is

$$\nabla \cdot \mathbf{u}^M = \nabla \cdot \mathbf{u}^V + [\rho]_\Gamma \left[\frac{1}{\rho}\right]_\Gamma \nabla \cdot \left(\frac{f(1-f)}{\rho^V} \dot{m} \mathbf{n}_\Gamma\right) \quad . \tag{7.15}$$

This expression can be further simplified by using $\nabla \cdot \mathbf{n}_\Gamma = \kappa$

$$\nabla \cdot \mathbf{u}^M = \nabla \cdot \mathbf{u}^V + [\rho]_\Gamma \left[\frac{1}{\rho}\right]_\Gamma \frac{f(1-f)}{\rho^V} \dot{m}\kappa \quad . \tag{7.16}$$

Then the corrected source term to impose is $\dot{s}_\rho = \dot{s}^V_\rho + \dot{s}^M_\rho$ with $\dot{s}^V$ the source term due to the volume source $\dot{s}^V_\rho = \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \mathbf{n}_\Gamma \cdot \nabla H_\Gamma$ and the source term due to mass source $\dot{s}^M_\rho = [\rho]_\Gamma \left[\frac{1}{\rho}\right]_\Gamma \frac{f(1-f)}{\rho^V} \dot{m}\kappa$.
A last approximation of $\dot{s}_\rho$ is proposed in [179] where geometric considerations are used. The expression relies on the wetted face areas A_f instead of the volume fraction f. This takes the form

$$\dot{s}_\rho = \dot{s}^V \frac{\rho_l \rho_g}{A_\Gamma} \left[\frac{(A^-_u - A^+_u)\, n_x}{\rho^-_u \rho^+_u} + \frac{(A^-_v A^+_v)\, n_y}{\rho^-_v \rho^+_v}\right] \quad , \tag{7.17}$$

with superscript $-$ and $+$ corresponding to the left and right faces respectively.
This new treatment showed a better behaviour of the Sherwood number evolution for an evaporating droplet compared to the iterative procedure of [195].
Note that this correction only needs to be done using a WDF, as the velocity has implicitly the meaning of a mass averaged velocity. This is not the case when using a JCF.

7.2.1.2 Jump condition formulation

The GFM applied to the JCF of the continuity equation was first introduced by [145] for the description of flame fronts and then applied by [218] for phase change computations.

Sharp ghost fluid method In this approach, the problem is not written as Eq. (7.7), instead of considering the velocity jump as a result of $\nabla \cdot \mathbf{u}^{n+1}$, it uses a different definition of $\nabla \cdot \mathbf{u}^*$ which takes into account the velocity jump

$$\nabla \cdot \mathbf{u}^* = \begin{cases} \nabla \cdot \mathbf{u}_g^* & \text{if } \phi < 0 \\ \nabla \cdot \mathbf{u}_l^* & \text{otherwise} \end{cases} \quad . \tag{7.18}$$

$\mathbf{u}_g$ and $\mathbf{u}_l$ are defined in their respective domain and in a narrow band of ghost cell in the other side of their domain.
Ghost values of these velocities take into account the velocity jump

$$\mathbf{u}_l^{gh} = \mathbf{u}_g + \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \mathbf{n}_\Gamma \quad , \tag{7.19}$$

$$\mathbf{u}_g^{gh} = \mathbf{u}_l + \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \mathbf{n}_\Gamma \quad , \tag{7.20}$$

with $\mathbf{u}_l^{gh}$ the ghost liquid velocity defined in Ω_g and $\mathbf{u}_g^{gh}$ the ghost gas velocity defined in Ω_l.
This requires to define $\dot{m}$ values at the velocity nodes usually done by linear interpolation.

Diffused ghost fluid method As discussed in Section 7.2.1.1, a sharp treatment can lead to stability problems and oscillatory behaviour for the pressure. This issue is addressed in [107] by modifying the ghost cell velocity definitions Eqs. (7.19) and (7.20). Instead of the sharp jump $\left[\frac{1}{\rho}\right]_\Gamma$, a smooth jump based on an inverse density β defined by

$$\beta = \frac{1}{\rho_l} H_\Gamma^\epsilon + \frac{1}{\rho_g}\left(1 - H_\Gamma^\epsilon\right) \quad , \tag{7.21}$$

with H_Γ^ϵ a smooth Heaviside defined as in Eq. (2.37). For a given cell, $\nabla\beta$ provides a smooth version of $\left[\frac{1}{\rho}\right]_\Gamma$ with a direct dependency on the interface subcell location.
Also, the linear interpolation of $\dot{m}$ to a velocity node is corrected by a curvature ratio

$$\dot{m}_{i-\frac{1}{2}} = \frac{\kappa_\Gamma}{\kappa_{i-\frac{1}{2}}} \dot{m}_\Gamma \quad , \tag{7.22}$$

with $\dot{m}_\Gamma$ the mass evaporation at the interface location obtained by extrapolation. In this correction, the curvature ratio stands for the difference of effective surface considering a curved interface located at $x_{i-\frac{1}{2}}$ instead of x_Γ.
Then, the ghost value of the gas velocity x-component $u_{g,i-\frac{1}{2}}^{gh}$ is

$$u_{g,i-\frac{1}{2}}^{gh} = u_{l,i-\frac{1}{2}} - \frac{\kappa_\Gamma}{\kappa_{i-\frac{1}{2}}} \dot{m}_\Gamma \left(\beta_{i-\frac{1}{2}} - \beta_i\right) n_{\Gamma,i} \quad . \tag{7.23}$$

Note that in this expression, the indices of β are not centred and depend of the node in which the operator $\nabla \cdot \mathbf{u}^*$ is evaluated. Here, it is evaluated in $\mathcal{C}_i$ so the β values are taken at $x_{i-\frac{1}{2}}$ and x_i. For $\mathcal{C}_{i-1}$, this would have been x_{i-1} and $x_{i-\frac{1}{2}}$.

7.2.1.3 General formulation of the velocity jump using WDF or JCF

In [107], authors pointed out that WDF and JCF belong to the same general framework where $\nabla \cdot \mathbf{u}^*$ is computed from the predicted one-fluid velocity $\mathbf{u}^*$ and $\nabla \cdot \mathbf{u}^{n+1}$ from

$$\boxed{\dot{s}_\rho = -\dot{m} \left[\frac{1}{\rho}\right]_\Gamma \mathbf{n}_\Gamma \cdot \nabla H_\Gamma} \tag{7.24}$$

In a continuous sense, the equivalence is straightforward by taking the scalar product of Eq. (5.37) and $\mathbf{n}_\Gamma$

$$\mathbf{n}_\Gamma \cdot \nabla H_\Gamma = \delta_\Gamma \quad . \tag{7.25}$$

Then Eq. (7.8) gives directly the relation of Eq. (7.24).
The choice of a velocity jump treatment in the continuity equation results in the choice of the source term $\dot{s}_\rho$ discretization. In Section 8.1, a comparison of different $\dot{s}_\rho$ discretizations is provided using the global metric of area error leading to the best-suited choices for both VOF and SLS.
Furthermore, the choice has to be consistent with the velocity representation.
If the velocity comes from a WDF, additional correction has to be used to ensure a mass-average divergence instead of the volume average difference presented here. This point is illustrated in the results of Section 9.3.

7.2.2 Velocity jump in the momentum equation

Depending of the formulation based either on WDF or JCF, the prediction step Eq. (5.1) has to take into account the velocity jump. In the derivation of the WDF presented in Chapter 2, the velocity jump is contained in the surface tension and does not need to be treated explicitly in $\mathcal{L}_{\text{conv}}$ or $\mathcal{L}_{\text{diff}}$. Thus, any of the schemes presented in Section 5.2 can be reused in this framework to compute $\mathcal{L}_{\text{conv}}$ and the viscosity has to be treated with a WDF accordingly.
However, some implementations [145; 218] uses a JCF which has to treat explicitly the velocity jump in the convection term. The $\mathcal{L}_{\text{conv}}$ operator is split into a liquid $\mathcal{L}_{\text{conv},l} = \rho_l \mathbf{u}_l \cdot \nabla \mathbf{u}_l$ and a gas operator $\mathcal{L}_{\text{conv},g} = \rho_g \mathbf{u}_g \cdot \nabla \mathbf{u}_g$ using only values $\mathbf{u}_l$ or $\mathbf{u}_g$ respectively. When the stencil crosses the interface, a simple GFM is applied to define ghost values from Eqs. (7.19) and (7.20). The $\mathcal{L}_{\text{conv}}$ is then defined as

$$\mathcal{L}_{\text{conv}} = \begin{cases} \rho_g \mathbf{u}_g \cdot \nabla \mathbf{u}_g & \text{if } \phi < 0 \\ \rho_l \mathbf{u}_l \cdot \nabla \mathbf{u}_l & \text{otherwise} \end{cases} \quad . \tag{7.26}$$

For the discretization of $\mathcal{L}_{\text{diff}}$, GFM can be used to impose $[\mathbf{u}]_\Gamma$ in the Laplacian analogously to what is done for the pressure jump in the Poisson equation. It can be treated as an explicit JCF as in [218] or HWDJCF as in [183]. A recent paper [163] also proposes an extension of implicit JCF to phase change applications.
In the unified framework, the choice has been made to use a momentum correction in $\mathcal{L}_{\text{conv}}$ and a WDF for $\mathcal{L}_{\text{visc}}$. Then, these choices are kept for the phase change solver as they can be easily extended without further treatment for the velocity jump.

7.3 Interface regression

The interface velocity needs to be modified to take into the phase change. This interface velocity is not divergence-free $\nabla \cdot \mathbf{u}_\Gamma \neq 0$ and the transport equation have to be reformulated into its initial advection form

$$\frac{\partial c}{\partial t} + \mathbf{u}_\Gamma \cdot \nabla c = 0 \quad , \tag{7.27}$$

or as,

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}_\Gamma c) = c \nabla \cdot \mathbf{u}_\Gamma \quad . \tag{7.28}$$

With Eq. (7.28), the natural form of the VOF transport arising from the WDF and Eq. (7.27) the natural form of the SLS transport.
From mass conservation in presence of phase change, the interface velocity can be expressed as

$$\mathbf{u}_\Gamma = \mathbf{u}_l - \frac{\dot{m}}{\rho_l}\mathbf{n}_\Gamma \quad , \tag{7.29}$$

$$\mathbf{u}_\Gamma = \mathbf{u}_g - \frac{\dot{m}}{\rho_g}\mathbf{n}_\Gamma \quad . \tag{7.30}$$

However, the interface velocity needs to be defined in the vicinity of the interface to perform a proper transport of c in the whole domain and the expressions above are not equivalent discretely at an arbitrary position. This is purely a numerical way around as $\mathbf{u}_\Gamma$ only have a definition at the interface position where Eqs. (7.29) and (7.30) are simultaneously true.
A natural way to define the interface velocity numerically is to rely on the WDF of velocity. In a Front Tracking framework [58] the interface velocity is retrieved by averaging the two expressions in Eqs. (7.29) and (7.30):

$$\mathbf{u}_\Gamma = \frac{1}{2}\left(\mathbf{u}_l + \mathbf{u}_g\right) - \frac{1}{2}\left(\frac{\dot{m}}{\rho_l} + \frac{\dot{m}}{\rho_g}\right)\mathbf{n}_\Gamma \quad . \tag{7.31}$$

This expression is also used in [51] with a VOF framework. As $\frac{1}{2}\left(\mathbf{u}_l + \mathbf{u}_g\right)$ corresponds to the volume average velocity $\mathbf{u}^V$ with $f = 0.5$, Eq. (7.31) only holds at the isocontour $f^{-1}(0.5)$. This is a natural definition for Front-Tracking where the velocity is defined at the markers which all belongs to the interface. However, in a VOF framework, $\mathbf{u}_\Gamma$ also needs to be defined at the interface location. In the implementation of [51], $\frac{1}{2}\left(\mathbf{u}_l + \mathbf{u}_g\right)$ is computed as a linear interpolation of the one-fluid velocity $\mathbf{u}$ at the PLIC interface centroid.
For the Level Set framework, Lee et al. [107] deduces the interface velocity from the one-fluid velocity $\mathbf{u}$ as

$$\mathbf{u}_\Gamma = \mathbf{u} + \dot{m}\beta\mathbf{n}_\Gamma \quad , \tag{7.32}$$

with β the diffused inverse density of Eq. (7.21). It can be seen as the generalization of Eq. (7.31) using the same volume averaged velocity assumption with the node of velocity located at an arbitrary isocontour value between 0 and 1. The general formulation is then

$$\boxed{\mathbf{u}_\Gamma = \left(\mathbf{u}_l - \frac{\dot{m}}{\rho_l}\mathbf{n}_\Gamma\right) H_\Gamma + \left(\mathbf{u}_g - \frac{\dot{m}}{\rho_g}\mathbf{n}_\Gamma\right)(1 - H_\Gamma) \quad .} \tag{7.33}$$

In [7], the expressions Eqs. (7.29) and (7.30) are switched with respect to the domain in which the velocity is defined. Thus it relies on the liquid velocity in Ω_l and on the gas velocity in Ω_g which is equivalent to Eq. (7.33) with $H_\Gamma = H_\Gamma^0$. This corresponds to the JCF formulation of the interface velocity.
Another approach considers only one of the two expressions. This reduces to Eq. (7.33) with $H_\Gamma = 0$ or $H_\Gamma = 1$ everywhere. Most of the time, this ad-hoc choice depends on the target application. While solvers for boiling applications [219; 236; 187] tend to rely on the gas velocity Eq. (7.30), solvers for evaporation often use the liquid velocity Eq. (7.29) instead [183; 192; 159]. As the interface velocity is needed in the whole domain (or at least in a narrow band around the interface) to perform the correct transport of the color function c, the phase velocity can be required everywhere depending on the choice of H_Γ in Eq. (7.33).

7.3.1 Phase velocities

In the above general expression of interface velocity Eq. (7.33), the velocity are separated as liquid and gas velocities. Depending on the choice of H_Γ, the phase velocity reconstruction requires different numerical treatment.

7.3.1.1 Whole domain formulation

If H_Γ is taken as H_Γ^ϵ, then it is consistent with the WDF of the velocity. Then the phase velocities are retrieved using the definition of Chapter 2 along with Eq. (7.6) as in the VOF formulation provided in [122]

$$\mathbf{u}_l = \mathbf{u}^V + (1 - f)\,[\mathbf{u}]_\Gamma \quad , \tag{7.34}$$

$$\mathbf{u}_g = \mathbf{u}^V - f\,[\mathbf{u}]_\Gamma \quad , \tag{7.35}$$

with $[\mathbf{u}]_\Gamma = \dot{m}\left[\frac{1}{\rho}\right]_\Gamma \mathbf{n}_\Gamma$.

7.3.1.2 Jump condition formulation

If H_Γ^0 is chosen, then the JCF of velocity is used and velocity can be picked safely as either liquid or gas depending on the domain in which it is defined without further treatment. The phase velocity can also be defined in the other phase using GFM proposed in [145; 218] through the use of Eqs. (7.19) and (7.20). It can be shown that GFM is discretely

equivalent to the JCF used in [7]. Let us take the case of a liquid formulation

$$\mathbf{u}_\Gamma = \mathbf{u}_l - \frac{\dot{m}}{\rho_l}\mathbf{n}_\Gamma \quad . \tag{7.36}$$

In the liquid phase ($H_\Gamma = 1$), this expression is totally defined while in the gas ($H_\Gamma = 0$), $\mathbf{u}_l$ is not defined. Using Eq. (7.19) the interface velocity is written as

$$\begin{aligned}\mathbf{u}_\Gamma &= \mathbf{u}_l^{gh} - \frac{\dot{m}}{\rho_l}\mathbf{n}_\Gamma = \mathbf{u}_g + \dot{m}\left[\frac{1}{\rho}\right]_\Gamma \mathbf{n}_\Gamma - \frac{\dot{m}}{\rho_l}\mathbf{n}_\Gamma \\ &= \mathbf{u}_g + \left(\frac{\dot{m}}{\rho_l} - \frac{\dot{m}}{\rho_g}\right)\mathbf{n}_\Gamma - \frac{\dot{m}}{\rho_l}\mathbf{n}_\Gamma \\ &= \mathbf{u}_g - \frac{\dot{m}}{\rho_g}\mathbf{n}_\Gamma \quad ,\end{aligned} \tag{7.37}$$

which corresponds exactly to Eq. (7.33) with $H_\Gamma = H_\Gamma^0$. The same can be shown from a gas definition of $\mathbf{u}_\Gamma$.

7.3.1.3 Velocity divergence-free extension

Finally, a phase velocity could be needed in the whole domain if H_Γ is taken arbitrarily equal to 0 or 1. It is then required to define extension of $\mathbf{u}_l$ and $\mathbf{u}_g$ in Ω_g and Ω_l respectively for the velocity to be continuous in whole domain. This can be done with the GFM extension presented above of using the PDE extrapolation presented in Appendix B.1 as in [124; 159; 187]. While this approach is purely numerical and does not prescribe the correct velocity jump between ghost and real cells, it ensures continuous velocity profiles across the interface.
The above method provide a velocity definition in the whole domain. However, the obtained velocity fields $\mathbf{u}_l$ and $\mathbf{u}_g$ are continuous but not divergence-free. This divergence-free error can lead to erroneous transport of the interface which directly impact the prediction of interface regression. The effect is more important in evaporation configurations [218] compared to boiling simulations [219]. This issue has first been addressed in [218] where successive projection steps are proposed to retrieve a divergence-free velocity field:

1. Extend the predicted liquid velocity $\mathbf{u}_l^*$ in the gas using Eq. (7.19).
2. Compute a ghost pressure P^{gh} using the following Poisson equation

$$\nabla \cdot \left(\frac{1}{\rho}\nabla P^{gh}\right) = \frac{1}{\Delta t}\nabla \cdot \mathbf{u}_l^* \quad . \tag{7.38}$$

3. Correct the velocity to obtain $\mathbf{u}_l^{**}$

$$\mathbf{u}_l^{**} = \begin{cases} \mathbf{u}_l & \text{in } \Omega_l \\ \mathbf{u}_l^* - \frac{\Delta t}{\rho}\nabla P^{gh} & \text{in } \Omega_g \end{cases} \quad . \tag{7.39}$$

4. $\mathbf{u}_l^{**}$ is discontinuous at the interface and so is not divergence free at this location. A second projection is then required to obtain the divergence-free liquid velocity $\mathbf{u}_l$. This is done by defining a potential Ψ following the Poisson equation

$$\Delta\Psi = \nabla \cdot \mathbf{u}_l^{**} \quad . \tag{7.40}$$

5. Finally retrieve $\mathbf{u}_l$ by correction

$$\mathbf{u}_l = \mathbf{u}_l^{**} - \nabla\Psi \quad . \tag{7.41}$$

This procedure requires to extend the velocity field $\mathbf{u}_l^*$ and then solve two additional elliptic equations to retrieve a divergence-free velocity $\mathbf{u}_l$ which is a huge increase in computational time.
In [183] a modified algorithm is used where step 4 and 5 are skipped for efficiency improvement and the liquid velocity is directly taken as $\mathbf{u}_l = \mathbf{u}_l^{**}$. This choice does not have a huge impact on the interface transport as the divergence-free error is concentrated at the interface location only.
Another method proposed in [159] take directly $\mathbf{u}_l$ at a given time step and project it on a divergence-free solution by using the following algorithm

1. Extend the liquid velocity $\mathbf{u}_l$ in the gas using a constant PDE extrapolation to retrieve $\mathbf{u}_l^*$.
2. Derive the potential Ψ by solving the Helmotz equation

$$\Delta\Psi + \omega\Psi = \nabla \cdot \mathbf{u}_l^* \quad , \tag{7.42}$$

   with ω a relaxation term which is zero in a narrow band around the interface and $\omega = 1/\Delta t^2$ otherwise.
3. Finally retrieve $\mathbf{u}_l$ by correcting it

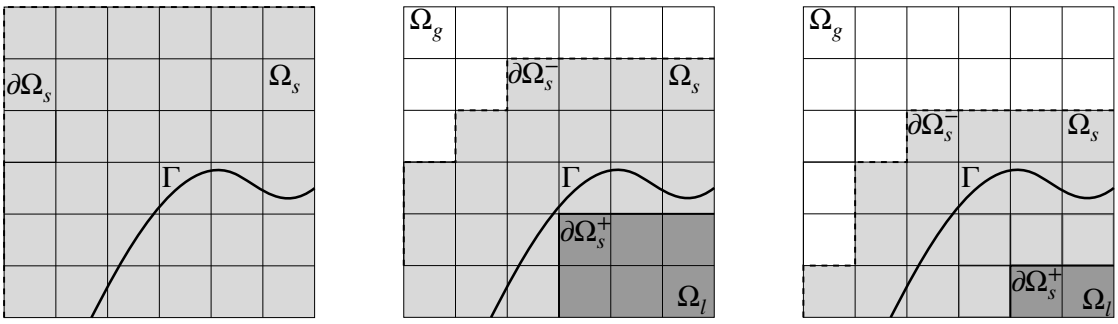
$$\mathbf{u}_l = \mathbf{u}_l^* - \nabla\Psi \quad . \tag{7.43}$$

The relaxation term allows to converge more quickly by alleviating the condition $\int_\Omega (\nabla \cdot \mathbf{u}_l^*)\, d\Omega = 0$. However, the divergence-free condition only holds in the cells where $\omega = 0$.

7.3.1.4 Stefan flow cancellation

Again based on the choice of H_Γ equal to 0 or 1, another way to define a continuous and divergence-free phase velocity field across the interface relies on the cancellation of the Stefan flow produced by the velocity jump.
As presented in [218], the velocity field related to the Stefan flow derives from the potential Ψ computed from the Poisson equation

$$\Delta\Psi = 0 \quad , \tag{7.44}$$



(a) *Scapin et al.* **(b)** *Malan et al.* **(c)** *Bures et al.*

Figure 7.1: *Illustration of the Stefan flow potential domain Ω_s (in light grey), boundaries $\partial\Omega_s$ (in dashed line), $\partial\Omega_s^-$ (in dashed line) and $\partial\Omega_s^+$ (in solid line)*

with the jump condition $[\nabla\Psi \cdot \mathbf{n}_\Gamma]_\Gamma = \dot{m}\left[\frac{1}{\rho}\right]_\Gamma$.
The Stefan flow velocity $\mathbf{u}_s$ is retrieved through the following projection in Ω

$$\begin{cases} \Delta\Psi = \dot{s}_\rho \\ \mathbf{u}_s = \nabla\Psi \end{cases} , \tag{7.45}$$

with boundary conditions $\partial\Omega_s$ taken as free outlet. However, in Eq. (7.45), the density is not taken into account.

Malan et al. approach In [123], the following density variable projection is performed only in the subdomain Ω_s

$$\begin{cases} \nabla \cdot \left(\frac{1}{\rho}\nabla\Psi\right) = \dot{s}_\rho \\ \mathbf{u}_s = \frac{1}{\rho}\nabla\Psi \end{cases} , \tag{7.46}$$

where $\Omega_s = \Omega_g^2 \cup \Omega_\Gamma$ with Ω_g^2 the two layers of gas cells adjacent to the interface (see Fig. 7.1b). No-slip boundary conditions are applied to $\partial\Omega_s^+ = \Omega_\Gamma \cap \Omega_l$ and free outlet boundary conditions are applied to $\partial\Omega_s^- = \Omega_g^2 \cap \left(\Omega_g \setminus \Omega^2\right)$.
The method cancels the Stefan flow velocity from the one-fluid velocity $\mathbf{u}$ to remove the velocity jump at the interface and obtain a velocity $\mathbf{u}_l$ at the interface

$$\mathbf{u}_l = \mathbf{u} - \mathbf{u}_s \quad . \tag{7.47}$$

It is easy to show that using either Eq. (7.45) or Eq. (7.46) for retrieving $\mathbf{u}_s$ leads to $\nabla \cdot \mathbf{u}_l = 0$ by noticing that

$$\nabla \cdot \mathbf{u}_l = \nabla \cdot \mathbf{u} - \nabla \cdot \mathbf{u}_s = \dot{s}_\rho - \dot{s}_\rho = 0 \quad . \tag{7.48}$$

Then the outcome liquid velocity is defined in $\Omega_l \cup \Omega_\Gamma \cup \Omega_g^2$ which is enough in the VOF context for proper advection of f.

Scapin et al. approach In [192], Eq. (7.45) is preferred as it allows the use of Fast Fourier Transform (FFT) direct solvers which is not possible for variable coefficient Poisson equation. The velocity $\mathbf{u}_l$ is not really the liquid velocity but more a one-fluid velocity without velocity jump. However, in an evaporation process, this can be used as an extended liquid velocity because the removed Stefan flow is in the gas domain.

Bures et al. approach In [24], Eq. (7.45) is also used but only in the restricted domain $\Omega_s = \Omega_g^1 \cup \Omega_\Gamma \cup \Omega_l^1$ illustrated in Fig. 7.1c with the same boundary conditions as for the approach in [123]. With this new domain definition, liquid velocity at faces of a mixed cells belonging to the liquid are recalculated because they are not enough reliable from the one-fluid velocity $\mathbf{u}$.
Note that in those elliptic problems, the use of free-outlet boundary conditions are mandatory for well-posedness of the projection method in the context of incompressible phase change as no dilatation is allowed in the gas.

7.3.1.5 Conclusion

While WDF and JCF relies on the derivation of the equations, the velocity divergence-free extension and the Stefan flow cancellation are numerical tools which allow to have the wanted properties of the velocity field for a robust transport of the interface. The Stefan flow cancellation constitutes a straightforward and generic way to obtain a divergence-free and continuous velocity field from any formulation used for velocity transport. From this perspective, it is the retained method for the present solver.

7.3.2 Phase change term

Once the phase velocities are properly defined in the domain, the regression term due to phase change $-\dot{m}/\rho_p$ needs to be treated carefully. This can be done either by solving the velocity form Eq. (7.28) with the interface velocity definition given by Eq. (7.33) or by imposing the regression as a source term.

7.3.2.1 Velocity form

The velocity form

$$\frac{\partial c}{\partial t} + \mathbf{u}_\Gamma \cdot \nabla c = 0 \quad , \tag{7.49}$$

gives a straightforward approach and has been widely used in the literature for Level Set framework [183; 7; 29] and more recently VOF framework [192; 159].
Using the velocity form, the interface velocity needs to be defined in the whole domain or at least in a narrow band around the interface. This requires to extend $\dot{m}$ away from the interface.
In the VOF framework, this extension is only required in cells adjacent to the mixed cells Ω_l^1 and Ω_g^1 while Level Set needs this in at least a narrow band around the interface.

From extrapolated fields In the early Level Set approaches [218], the evaporation rate was not extrapolated but directly computed from $\mathcal{M}_T$ or $\mathcal{M}_Y$ in the whole domain. In [218], the field Y is extrapolated linearly using the PDE approach of Appendix B.1.1. Then, the evaporation rate $\dot{m}$ is retrieved using $\mathcal{M}_Y$ with $Y_\Gamma = Y$ the local species value. Note that this method leads to non-physical values of $\dot{m}$ far from the interface. In fact, the accuracy of this term is not important because the Level Set is reinitialized after the transport. However computing the evaporation rate in the whole domain provides a continuous $\dot{m}$ which is a very important feature for stable transport of ϕ.
In [159], the evaporation rate is computed in Ω_l^1 and Ω_g^1 (one layer around the interface) by using the Least Square method based on $\mathcal{M}_T$ and the fields T_l and T_g extrapolated linearly with the PDE approach presented in Appendix B.1.1. The computation is close enough to the interface to remain accurate as the stencil still implies interface cells.

From extrapolated gradients In [33], $\dot{m}$ is defined in a narrow band across the interface by computing the gradients ∇T_l only in $\Omega_l \cap \Omega_\Gamma$ and extrapolated in a constant fashion in Ω_g, the same is done for ∇T_g. This gives a natural definition of $\dot{m}$ by using $\mathcal{M}_T$. This approach as also been used in [7; 24].

From extrapolated evaporation rate In [219] proposed an improvement of the above method by directly extrapolating the evaporation rate from $\Omega_l \cap \Omega_\Gamma$ to $\Omega_g \setminus \Omega_\Gamma$ and from $\Omega_g \cap \Omega_\Gamma$ to $\Omega_l \setminus \Omega_\Gamma$ This is still done using the PDE approach of Appendix B.1.1. Such approach allows to have a constant $\dot{m}$ value in the normal direction to the interface which is more accurate than computing it from extrapolated fields.
In [7] authors argue that the extrapolation of $\dot{m}$ should be from $\Omega_l \cap \Omega_\Gamma$ to Ω_l and from $\Omega_l \cap \Omega_\Gamma$ to Ω_l instead of the initial procedure of [219]. If the temperature is maintained at the interface temperature, both procedures are equivalent. However, when both ∇T_l and ∇T_g are non-zero, the evaluation of $\dot{m}$ is only valid in its side of the interface.

7.3.2.2 Source term

The source term approach is based on the WDF applied to c leading to the following equation

$$\frac{\partial c}{\partial t} + \nabla \cdot (\mathbf{u}_p c) = \frac{\dot{m}}{\rho_p} |\nabla c| \quad , \tag{7.50}$$

with the subscript $p = l, g$ referring to a phase.
In the context of Level Set, this expression has been used in [187] as

$$\frac{\partial c}{\partial t} + \mathbf{u}_g \cdot \nabla \phi = \frac{\dot{m}}{\rho_g} \quad , \tag{7.51}$$

with $|\nabla \phi| = 1$ by definition. This method, as for the velocity form, needs an extension of $\dot{m}$ away from the interface. This is done using one of the approaches detailed in Section 7.3.2.1.

In the VOF framework, several authors [195; 190; 236; 192; 123] proposed the following formulation for $c = f$

$$\frac{\partial c}{\partial t} + \mathbf{u}_l \cdot \nabla \phi = \frac{\ddot{m}}{\rho_l} \quad , \tag{7.52}$$

with $\ddot{m}$ the volumetric mass evaporation rate such that

$$\ddot{m} = \dot{m} \frac{A_\Gamma}{V_\mathcal{C}} \quad . \tag{7.53}$$

For geometric algorithms of VOF, this source term step is performed after the transport step in order to avoid any overshoot or undershoot. Moreover, this source term does not require extension of the evaporation rate as it is naturally located in mixed cells.

Algebraic approach In [195], the mass source $\frac{\ddot{m}}{\rho_l}$ is simply added to the volume fraction field after advection f^* and limited such that

$$\frac{\ddot{m}}{\rho_l} \leq \frac{f^*}{\Delta t} \quad . \tag{7.54}$$

This methodology is robust but can suffer from mass conservation if the clipped values are not redistributed.

Geometric approach In [123], the source term is applied by using the PLIC reconstruction. Instead of applying a mass source in the control volume, the source term is reformulated into a shift of the plane parameter. The mass change is then evaluated by evaluating the new PLIC reconstruction. The following algorithm is used for that purpose

1. Compute the plane parameter shift Δd such that

$$\Delta d = -\frac{\dot{m}}{\rho_l} \Delta t \quad . \tag{7.55}$$

2. Update the plane parameter as

$$d^{n+1} = d^* + \Delta d \quad , \tag{7.56}$$

   with d^{n+1} the plane parameter of the PLIC reconstruction based on f^{n+1} and d^* the plane parameter of the PLIC reconstruction based on the volume fraction after advection f^*.
3. Check for potential overshoots/undershoots by computing d_{min} and d_{max} the plane parameter corresponding to the PLIC reconstruction from $f = 0$ and $\mathbf{n}^*$ and $f = 1$ and $\mathbf{n}^*$ respectively.
   - If $d_{\min} < d^{n+1} < d_{\max}$ then simply compute the new volume fraction f^{n+1} from d^{n+1} and $\mathbf{n}^*$.
   - If $d^{n+1} \leq d_{\min}$ then there is undershoot, $f^{n+1} = 0$ and mark the cell a drain cell D.

- If $d_{\max} \geq d^{n+1}$ then there is overshoot, $f^{n+1} = 1$ and mark the the cell as a source cell S.

4. Loop over all cells to redistribute the overshoot of S cells to empty neighbouring cells and undershoot of D cells to full neighbouring cells.

This method is fully conservative as it redistributes the volume to other cells in case of clipping.
In [192], the use of a velocity form is found to be more accurate than the source term approach described above for the interface regression even if the VOF geometric methods are initially designed for divergence-free velocity field.

7.3.2.3 Conclusion

From the above state-of-the-art, the velocity form is employed here for both VOF and SLS as it is extended naturally from the capturing method algorithms presented in Chapter 5. Moreover, this equation form seems to provide the most accurate results in light of literature results. Hence, a proper extension of $\dot{m}$ is required in a narrow band around the interface which is performed by directly extrapolating constantly in the normal direction to the interface with the method of [7].

7.4 Scalar equations with flux jumps

Incompressible phase change simulations need to solve the energy equation

$$\frac{\partial \rho c_p T}{\partial t} + \nabla \cdot (\rho \mathbf{u} c_p T) = \nabla \cdot (k \nabla T) \quad , \tag{7.57}$$

with the associated flux jump

$$[k \nabla T \cdot \mathbf{n}_\Gamma]_\Gamma = \dot{m} \, [h]_\Gamma \quad , \tag{7.58}$$

and the species mass fraction equation

$$\frac{\partial \rho Y}{\partial t} + \nabla \cdot (\rho \mathbf{u} Y) = \nabla \cdot (\rho \mathcal{D}_v \nabla Y) \quad , \tag{7.59}$$

with the flux jump

$$\rho_g \mathcal{D}_v \nabla Y \cdot \mathbf{n}_\Gamma = \dot{m} \, (Y_\Gamma - 1) \quad . \tag{7.60}$$

In a more general way, they can be defined as general scalars Φ transported by the velocity $\mathbf{u}$ and diffuse with a diffusion coefficient β:

$$\frac{\partial \rho \Phi}{\partial t} + \nabla \cdot (\rho \mathbf{u} \Phi) = \nabla \cdot (\beta \nabla \Phi) \tag{7.61}$$

with the associated flux jump $[\beta \nabla \Phi \cdot \mathbf{n}_\Gamma]_\Gamma$ at the interface.
The transport of Φ has to be treated carefully to preserve accuracy of the solution even close to the interface. Indeed, the accuracy of temperature and species fields is of high

importance as it is used for the computation phase change quantities $\dot{m}$, T_Γ and Y_Γ defined in Section 7.1. The resolution of Eq. (7.61) can be done with the classic WDF used for velocity with the same numerical tools presented for continuity and momentum equations. However, this approach naturally smears the discontinuities over few cells at the interface and quantities are then mass-averaged (or enthalpy-averaged in the case of temperature in light of Eq. (2.28)). As the use of averages quantities is not suited for accurate computation of phase change quantities at the interface, another approach is to use the JCF or the TFF of Eq. (7.61) presented in Chapter 2. These approaches provide well-defined quantities of each phase even at the interface vicinity but bring new numerical challenges presented hereafter.

7.4.1 Whole domain formulation

This section briefly introduces the numerical implementation of a WDF for the energy equation Eq. (7.57) with an associated jump Eq. (7.58) and the species mass fraction Eq. (7.59) with the flux jump Eq. (7.60).

7.4.1.1 Convective and diffusive operators

The WDF, is mostly used with VOF [195; 103; 192; 52] or Front Tracking frameworks [58; 85] where $\mathcal{L}_{\text{conv}}(\rho\Phi)$ is computed as

$$\mathcal{L}_{\text{conv}}(\rho\Phi) = \nabla\cdot(\rho\mathbf{u}\Phi) \quad . \tag{7.62}$$

The algorithm proposed in [123] introduces an interesting feature for the fluxes $F^{(\rho\Phi)}$ computation. They are computed using the geometric flux $F^{(f)}$ to keep consistency between mass and scalar transport. This is analogous to the consistency between mass and momentum transport presented in Section 5.2. For the energy, this gives

$$F^{(\rho c_p T)} = \left(\rho_l c_{p,l} F^{(f)} + \rho_g c_{p,g} F^{(1-f)}\right)\widetilde{T} \quad , \tag{7.63}$$

with $\widetilde{T}$ the interpolated temperature at the cell face.
In [195; 122], Y is treated like a VOF variable $\mathcal{L}_{\text{conv}}(\rho Y)$ is computed from geometric fluxes. Then, the species mass fluxes are written as

$$F^{(\rho Y)} = \rho_g F^{(1-f)}\widetilde{Y} \quad , \tag{7.64}$$

with $\widetilde{Y}$ an interpolation of Y at the cell face.
In [85; 236; 192], a non-conservative form is used for T

$$\mathcal{L}_{\text{conv}}(\rho c_p T) = \mathbf{u}\rho c_p\cdot\nabla T \quad , \tag{7.65}$$

where ρc_p is considered as a quantity obtained through linear for [85] or harmonic [192] average based on H_Γ.

$$\rho c_p = \rho_l c_{p,l} H_\Gamma + \rho_g c_{p,g}(1-H_\Gamma) \qquad \text{for linear interpolation} \quad , \tag{7.66}$$

$$\rho c_p = \frac{\rho_l c_{p,l}\rho_g c_{p,g}}{\rho_g c_{p,g} H_\Gamma + \rho_l c_{p,l}(1-H_\Gamma)} \qquad \text{for harmonic interpolation} \quad . \tag{7.67}$$

While the linear average directly comes from the WDF of energy Eq. (2.28) where c_p is mass average, the harmonic average arises from a numerical choice.
The non-conservative form can also been used for Y:

$$\mathcal{L}_{\text{conv}}(\rho Y) = \mathbf{u}\rho \cdot \nabla Y \quad , \tag{7.68}$$

with $\rho = \rho_g$.
In [192], the conservative and non-conservative discretizations are compared on psychrometric data for a general droplet evaporation process. As expected, the conservative discretization provides better results for the coarse meshes while the difference is less significant at higher resolution. However, this test case has been performed with a density ratio of order 10 which is far from target applications of fuel droplets. The discrepancy produced by using such non-conservative form is then expected to be even more important for higher density ratios.
On the other hand, the diffusion operator $\mathcal{L}_{\text{diff}}$ is often treated with classical second-order operators.

7.4.1.2 Interface contribution

In the WDF, the interface contribution appears as a source term including the mass change contribution and the flux jump. It is either done with the smooth Dirac distribution Eq. (7.9) [192; 52], the sharp Dirac distribution Eq. (7.10) [195], or a smoothing of the sharp Dirac [103]. It has also been introduced in the form $\delta_\Gamma = \mathbf{n}_\Gamma \cdot \nabla H_\Gamma$ for the Level Set approach in [205] with $H_\Gamma = H^0_\Gamma$.
For the species mass fraction Y, a mass source due to phase change is accounted for by imposing the source term only in the gas part. Then, δ_Γ is discretized using Eq. (7.10) in [195; 122] which ensures that no source term is applied in the liquid part of the domain while an asymmetric Dirac distribution is used in [85].
Another approach proposed in [123] takes into account the flux jump by splitting the energy into a gas part and a liquid part and imposing a boundary conditions T_Γ in the diffusion operator. This is inspired from the JCF formulation presented hereafter and allows to take naturally the energy flux jump through the modification of $\mathcal{L}_{\text{diff}}$. The methods to impose boundary conditions at the interface are numerically challenging and are detailed in Section 8.3. In such case, only the flux jump is taken into account and the contribution due to mass change needs to be included as an additional change in the energy $\Delta f\,(c_{p,g} - c_{p,l})\,T_\Gamma$ with Δf the change of volume fraction due to phase change.

7.4.2 Jump condition formulation

The JCF was presented in [68] for Stefan problems and then extended to evaporation and boiling by several authors in Level Set framework [33; 219; 183; 107; 29; 7]. It has also been done using the VOF framework in [190; 236]. This leads to the following equation of temperature only defined in its respective phase p

$$\rho_p c_{p,p} \left(\frac{\partial T_p}{\partial t} + \mathbf{u}_p \cdot \nabla T_p \right) = \nabla \cdot (k_p \nabla T_p) \qquad \text{in } \Omega_p \quad . \tag{7.69}$$

For the species mass fraction, the liquid part of Eq. (7.59) reduces to $Y_{v,l} = 1$ while the gas part$Y_{v,g}$ (in the following the subscript g is dropped for clarity, then $Y_{v,g} = Y$) needs to be deduced from

$$\rho_g \left(\frac{\partial Y}{\partial t} + \mathbf{u}_g \cdot \nabla Y \right) = \nabla \cdot (\mathcal{D}_v \nabla Y) \qquad \text{in } \Omega_g \quad . \tag{7.70}$$

In these equations, no interface contribution appears explicitly and needs to be taken into account in the discretization operators near the interface. This can take the form of a flux jump or a boundary condition.
For the temperature

$$[k \nabla T \cdot \mathbf{n}_\Gamma]_\Gamma = \dot{m} \, [h]_\Gamma \qquad \text{at } \Gamma \quad , \tag{7.71}$$

$$T = T_\Gamma \qquad \text{at } \Gamma \quad , \tag{7.72}$$

or for the species

$$\rho_g \mathcal{D}_v \nabla Y \cdot \mathbf{n}_\Gamma = \dot{m} \, (Y_\Gamma - 1) \qquad \text{at } \Gamma \quad , \tag{7.73}$$

$$Y = Y_\Gamma \qquad \text{at } \Gamma \quad . \tag{7.74}$$

7.4.2.1 Convective operator

Using the JCF allows to split the convective operator $\mathcal{L}_{\text{conv}}(\rho\Phi)$ into a liquid $\mathcal{L}_{\text{conv}}(\rho_l \Phi_l)$ and a gas operator $\mathcal{L}_{\text{conv}}(\rho_g \Phi_g)$ only valid in there respective phase.
When the stencil crosses the interface, the scalar ghost cells Φ_l^{gh} and Φ_g^{gh} are computed using linear [218] or quadratic [219; 183] normal extrapolation (see Appendix B.1 for details). The use of quadratic extrapolation shows improvement in both boiling [219] and evaporation [183] test cases as it provides more accurate ghost cells. Note that this extrapolation procedure is employed instead of GFM because the temperature jump $[T]_\Gamma = 0$ does not allow to define well-suited ghost cells.
The $\mathcal{L}_{\text{conv}}$ is then defined as

$$\mathcal{L}_{\text{conv}}(\rho\Phi) = \begin{cases} \mathbf{u}_g \rho_g \cdot \nabla \Phi_g & \text{if } \phi < 0 \\ \mathbf{u}_l \rho_l \cdot \nabla \Phi_l & \text{otherwise} \end{cases} \quad , \tag{7.75}$$

with schemes of arbitrary order as ghost cells can be defined in a large stencil around the interface.
For a direct imposition of Eq. (7.72) in the convective operator, Sato et al. [190] and later Shaikh et al. [198] and Anumolu et al. [7] proposed to include T_Γ in the stencil through one-sided first-order finite differences instead of using ghost cell values. In the same idea, an Aslam-Chiu method was used in [21; 126] to prescribe Y_Γ explicitly at the interface for convective terms discretization.

7.4.2.2 Diffusive operator

For the diffusive operator, the GFM methodology of Appendix A.1 can be directly applied to the diffusion term by taking $a_\Gamma = 0$, $b_\Gamma = [\beta \nabla \Phi]_\Gamma$ and $S = \frac{\partial \rho \Phi}{\partial t} + \mathcal{L}_{\text{conv}}(\rho\Phi)$ as

proposed in [218; 198; 126]. Another approach is to use the ghost cell method presented in Section 8.3.2 to impose boundary conditions at the interface instead of $[\beta\nabla\Phi]_\Gamma$. The works in the literature [33; 218; 219; 183; 198; 107; 29; 123] often consider the second-order discretization to impose a Dirichlet boundary conditon Φ_Γ as it is straightforward to implement and lead to symmetric matrices easier to inverse. It is also possible to consider one-sided second-order differences for the discretization of $\mathcal{L}_{\text{diff}}$ in [190; 7] instead of the ghost cell method presented above. However, this requires larger stencil which are more difficult to treat implicitly. For the species, some authors [160; 183] choose to apply the Robin boundary condition Eq. (7.73) instead of the Dirichlet boundary condition Eq. (7.74) to prevent some numerical difficulties detailed in Section 9.2.1.3.

7.4.3 Two-fluid formulation

Finally, a two-fluid approach can be derived in the VOF framework as in [122; 159]. The set of equations is no longer Eq. (7.61) but

$$\rho_p \left(\frac{\partial \alpha_p \Phi_p}{\partial t} + \nabla \cdot (\alpha_p \mathbf{u}_p \Phi_p) \right) = \nabla \cdot (\alpha_p \beta_p \nabla \Phi_p) + \dot{\gamma}_p \delta_\Gamma \qquad \text{in } \Omega_p \quad , \tag{7.76}$$

with α_p the volume fraction of a given phase $p = l, g$ and $\dot{\gamma}_p$ the interface contribution related to phase p.

7.4.3.1 Convective and diffusive operator

The convective operator $\mathcal{L}_{\text{conv}}(\alpha_p \rho_p \Phi_p)$ also includes the volume fraction α_p. By noticing that $\alpha_l = f$ and $\alpha_g = 1 - f$, the fluxes $F^{(\alpha_p \rho_p \Phi_p)}$ are based on the fluxes F^f and F^{1-f} in [122].

$$F^{(\alpha_l \rho_l \Phi_l)} = \rho_l F^f \widetilde{\Phi}_l \quad , \tag{7.77}$$

with $\widetilde{\Phi}_l$ a face interpolation of Φ_l. This methodology ensures discrete consistency between mass and scalar transport.
Another approach based on the momentum algorithm of Section 5.2.3 was used in [159] where an additional continuity equation is solved for α_p and the fluxes of α_p are directly used in the scalar fluxes

$$F^{(\alpha_p \rho_p \Phi_p)} = \rho_p F^{\alpha_p} \widetilde{\Phi}_p \quad . \tag{7.78}$$

In [122] the diffusive operator $\mathcal{L}_{\text{diff}}$ is discretized with a standard second-order discretization. When the stencil implies Φ_p in the other phase, ghost cells are used.

7.4.3.2 Interface contribution

The interface contribution $\dot{\gamma}_p$ in the case the energy equation takes the form

$$\dot{\gamma}_p = \lambda_p \, \nabla T_p|_\Gamma \cdot \mathbf{n}_\Gamma + \frac{\dot{m}}{\rho_p} T_\Gamma \tag{7.79}$$

In [122], $\nabla T_p|_\Gamma$ and T_Γ are evaluated from a fitted temperature profile in the subcell described in Section 7.1. Another approach is used in [159] with $\dot{\gamma}_p$ imposed through the Dirichlet boundary condition T_Γ or Y_Γ depending on the considered equation. For this aim, the cut-cell method with phase barycentre values of Section 8.3.1 is applied. As for the JCF, this inherently includes the flux jump at the interface. However, the second term due to mass transfer does not seem to be taken into account in this procedure.

7.4.4 Conclusion

In this section, the different ways to solve the energy and species equations in a two-phase flow have been presented. The use of WDF can lead to unwanted mixing of values at the interface which makes difficult the computation of the interface quantities $\dot{m}$, T_Γ and Y_Γ. However, strategies are presented in Section 8.2 to circumvent this issue. The JCF allows to have a natural separation between phase quantities which is expected to improve the reconstruction of interface quantities. For the same reasons, the TFF seems to be a promising alternative which implies additional numerical challenges to transport both phase quantities.
When using WDF or TFF, the convective term has to be consistent with the interface transport in order to avoid any unphysical behaviours close to the interface. In the present solver, the consistent scalar transport of [159] is used for its natural way to handle this issue for any interface capturing method.
The handling of flux jumps at the interface have been presented can be treated either with a source term or by imposing a jump condition or a boundary condition at the interface. When jump or boundary conditions are imposed at the interface, the diffusion operator has to be treated accordingly. From the literature, a proper choice of flux jump handling is not clear as various approaches seem to provide encouraging results. Then it requires further numerical investigations detailed in Section 8.3 with a comparison between different methods.

7.5 Conclusion

This chapters details the different challenges associated with the integration of phase-change physics in an incompressible two-phase flow solver. It encompasses :

- Subcell thermodynamic closure at the interface to defined accurate interface quantities $\dot{m}$, T_Γ and Y_Γ.
- Velocity discontinuity appearing in both the continuity and momentum equation which needs careful implementation for stable computations.
- Interface regression due to phase change for accurate mass evolution.
- Sharp representation of temperature by solving the energy equation in presence of heat flux jump at the interface.
- Mass species production through phase change.

An overview of the literature on the subject has been presented with specific efforts given in the categorization of the different approaches of the literature and their link to WDF, JCF or TFF.

All these new challenges require very local reconstruction of quantities at the interface to prescribe the correct jump and boundary conditions which still needs to be determined. Th summary of choices made from this literature review are presented in Table 7.1 along with the points that still need to be clarified in the next chapters.

Numerical challenge	Choice in the solver
$\dot{m}$ evaluation	See Section 8.2
T_Γ and Y_Γ	Iterative process
Continuity	See Section 9.3.2
Momentum	WDF
Interface regression	Velocity form with $\mathbf{u}_l$
$\mathbf{u}_l$ construction	Stefan flow cancellation
Scalars	See Section 8.3

Table 7.1: *Summary of computation choices for phase-change challenges.*

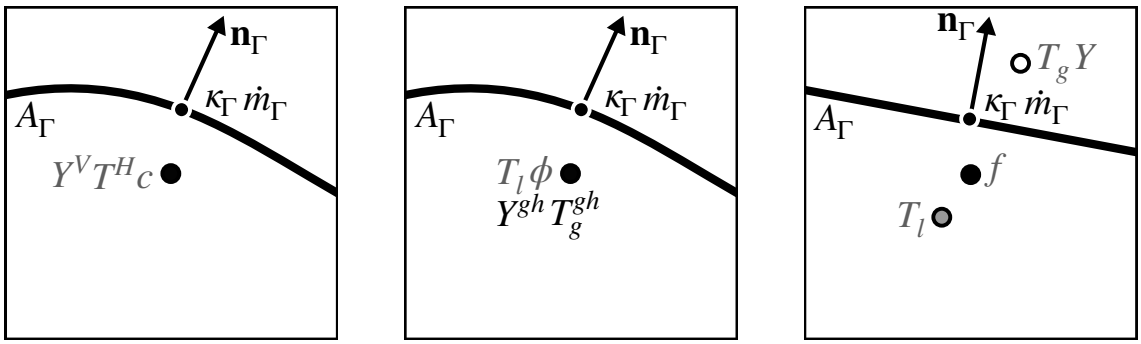
The next chapter proposes a complete study of numerical methods associated to the reconstructions at the interface as it is the key point of an accurate phase-change procedure. Then, the reconstructed quantities are used to impose the correct interface contributions using various methods presented in Section 7.4.

Chapter 8

Numerical reconstruction of interface quantities

Contents

In the previous chapter, the challenges of incompressible two-phase flow simulations with phase change have been exposed. They imply the reconstruction of additional quantities at the interface to handle the new discontinuities arising from the phase-change phenomenon. In this chapter, the goal is to give details on the reconstruction of these quantities.
Some aspects have already been mentioned for the reconstruction of normal in Section 3.1.1.1 and curvature in Section 5.4.2 which are based on c. Here, the scope is enlarged to encompass the reconstruction of the interface area and the evaporation rate at the interface as illustrated in Fig. 8.1. These new quantities are required to impose the correct flux jumps at the interface in presence of phase change.



(a) *WDF* **(b)** *JCF* **(c)** *TFF*

Figure 8.1: *Illustration of the reconstruction of interface quantities for WDF, JCF and TFF approaches. White points are the gas barycenters, grey points are the liquid barycenters and black points are the cell centers. Grey quantities are available through transport while black quantities have to be reconstructed.*

A first interrogation resides in the use of the same methodologies in a VOF and SLS framework to deal with the numerical challenges of phase-change simulations. It has been demonstrated in Chapter 6 that it is not a relevant choice for the curvature computation because of the difference of interface information provided by the interface capturing methods. Therefore, it is not straightforward to make a choice suited for all interface capturing methods without jeopardizing accuracy or consistency.
Using TFF, the temperature and species mass fraction are available at the phase barycenter (see Fig. 8.1c), which leads to a non-uniform discretization of operators while in a JCF framework, cells only contain the information of one phase and need a reconstruction strategy to obtain the other phase informations as illustrated in Fig. 8.1b. Finally, the WDF presented in Fig. 8.1a provides averaged quantities at the interface, which need to be used with care to avoid inaccuracies in the local reconstruction of phase gradients. This huge difference in the numerical representation may lead to drastically different ways of reconstructing $\dot{m}$.
Hence, this section tries to answer the following questions:

- Is it possible and relevant to apply exactly the same strategies to handle phase change for VOF and SLS ?
- For each reconstruction quantity, what is the best balance between accuracy, efficiency and implementation effort ?
- If each interface representation uses their most suited methods, which one provides the most accurate phase-change modelling ?

In Section 8.1, the reconstruction of interface area and Dirac distributions is investigated with an overview of the literature and a comparison of area accuracy for the different strategies. Then, the reconstruction of $\dot{m}$, T_Γ and Y_Γ is investigated in Section 8.2. The main discussion is about the reconstruction of accurate gradients at the interface to compute $\dot{m}$ based on the available fields. Finally, Section 8.3 presents the methods to handle flux jumps at the interface using either a VOF or a SLS representation. WDF, JCP and TFF

applied to VOF and SLS are compared on a static diffusion problem analogous to the heat or mass species fraction transfer equations.

8.1 Interface area and Dirac distribution

For phase-change simulations, the interface area is used to derive a collocated Dirac distribution for applying a local source term related to surface contributions. As presented in Chapter 7, this source term appears in the continuity, energy and species mass fraction equations if they are treated with a WDF. It is then really important to investigate the accuracy of such Dirac distribution.
If instead, the TFF is used, an interface reconstruction is needed to apply embedded boundary conditions using the cut-cell method presented in Section 8.3.1. This implies the knowledge of the interface area A_Γ, the wetted areas A_f and the volume fraction f in a given computational cell. In a VOF framework, the natural interface reconstruction is the PLIC used for geometric advection. However, it is interesting to consider other reconstruction procedures applied to a VOF framework for a potential accuracy improvement on the evaluation of A_Γ. In a SLS framework, no reconstruction is inherently available whereas it is needed if a cut-cell method is used to apply embedded boundaries.
In this section, different area reconstructions are presented and compared through numerical experiments. The same 2D test case as in Section 6.2.1.1 is considered where area and volume of a circle are computed. The related errors are defined as

$$E_{area} = \frac{|A_{exact} - A_{reco}|}{A_{exact}} \quad , \tag{8.1}$$

$$E_{volume} = \frac{|V_{exact} - V_{reco}|}{V_{exact}} \quad , \tag{8.2}$$

with $A_{exact} = 2\pi R$ and $V_{exact} = \pi R^2$ the exact area and volume respectively. The area and volume of the reconstruction are computed as

$$A_{reco} = \sum_{i=1}^{N_\mathcal{C}} A_{\Gamma,i} \quad , \tag{8.3}$$

$$V_{reco} = \sum_{i=1}^{N_\mathcal{C}} f_i \mathcal{V}_i \quad , \tag{8.4}$$

with $A_{\Gamma,i}$ the area of the reconstructed interface and f_i the fraction of liquid delimited by the reconstructed interface.
For a range of $N_D = 3.2$ to $N_D = 409.6$, the errors are evaluated on 100 circles randomly located in the domain in order to meet as much configurations as possible. The final metrics are $\langle E_{area} \rangle$ and $\langle E_{volume} \rangle$ the mean of E_{area} and E_{volume}, respectively, over the 100 circles.

8.1.1 Algebraic formulae

In the VOF framework, direct differentiation of the volume fraction f can provide the interface surface. The most popular formula [15; 207] was used in the context of algebraic

VOF where a PLIC reconstruction is not available for topology calculation.

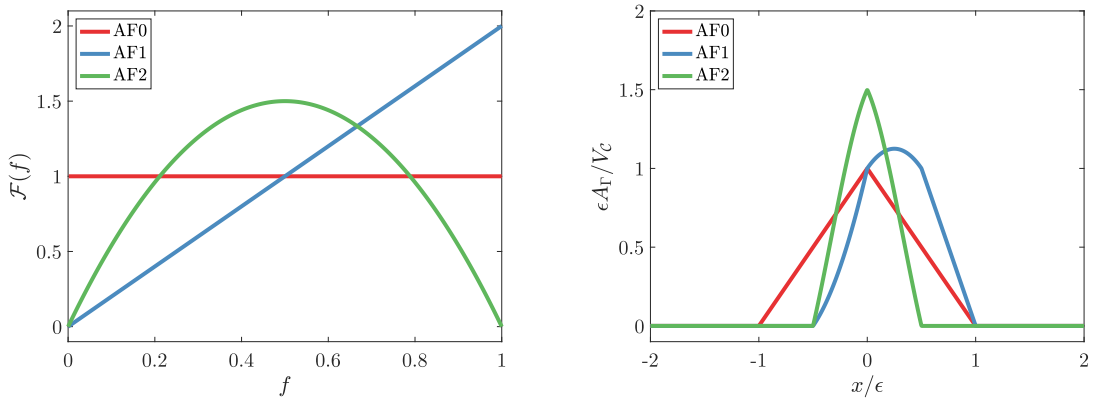
$$A_\Gamma = |\nabla f| V_\mathcal{C} \quad , \tag{8.5}$$

with $V_\mathcal{C}$ the cell volume. The issue of such approximation is that it is a global formulation and can suffer from local inaccuracies. Moreover, it provides finite values of interface area even in cells which does not contain the interface. This is not compatible with a sharp representation of the interface and can lead to shape deformations through phase change [202].

A more general formula was proposed in [75] to sharpen the interface. The idea is to derive formulae of the form $\mathscr{F}(f)|\nabla f|$ such that $\int_0^1 \mathscr{F}(f)df = 1$. Taking $\mathscr{F}(f) = 1$ leads to Eq. (8.5) while two other choices $\mathscr{F}(f) = 2f$ and $\mathscr{F}(f) = 6f(1-f)$ provide sharper formulations. This leads to three formulae:

- AF0 the formula $|\nabla f|$
- AF1 the formula $2f|\nabla f|$
- AF2 the formula $6f(1-f)|\nabla f|$

For clarity, the functions $\mathscr{F}(f)$ are plotted in Fig. 8.2a while the evaluation of $A_\Gamma V_\mathcal{C}$ is represented in Fig. 8.2b with respect to x/ϵ with ϵ the interface width Δx in 1D. Both AF1 and AF2 provide sharper formulation. However AF1 is anisotropic and does not lead to a sharpening in the liquid side (where $f > 0$).



(a) *Representation of* $\mathscr{F}(f)$ **(b)** *Evaluation of* $\epsilon A_\Gamma / V_\mathcal{C}$

Figure 8.2: *Illustration of different algebraic formulations for the interface area*

Even if those formulae have been originally introduced in the VOF framework, it is also possible to extend the algebraic method to any Heaviside H_Γ. In this study, algebraic formulae applied to both $H_\Gamma^\epsilon(f)$ and $H_\Gamma^\epsilon(\phi)$ defined as

$$H_\Gamma^\epsilon(f) = f \quad , \qquad H_\Gamma^\epsilon(\phi) = \frac{1}{2}\left(\tanh\left(\frac{\phi}{2\epsilon}\right)\right) \quad , \tag{8.6}$$

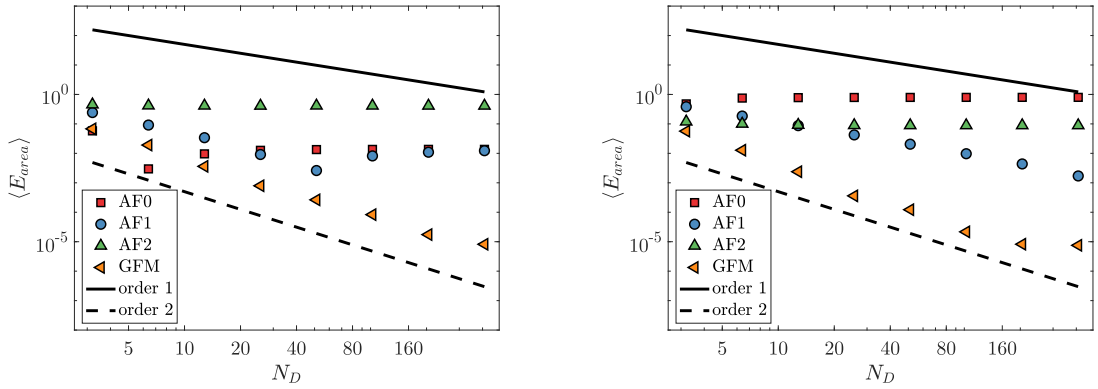
are investigated along with the alternative formula used in the GFM framework based on H_Γ^0:

$$A_\Gamma = \mathbf{n}_\Gamma \cdot \nabla H_\Gamma^0 V_\mathcal{C} \quad , \tag{8.7}$$

with H^0_Γ defined at the faces using the faces values. This gives in 1D

$$A_{\Gamma,i} = \left(H^0_\Gamma \left(\frac{1}{2}(c_{i+1} + c_i) \right) - H^0_\Gamma \left(\frac{1}{2}(c_i + c_{i-1}) \right) \right) \frac{n_{x,i}}{\Delta x} V_{\mathcal{C},i} \quad , \tag{8.8}$$

with $H^0_\Gamma(c) = 1$ in the liquid and $H^0_\Gamma(c) = 0$ in the gas.



(a) *From f* **(b)** *From ϕ*

Figure 8.3: *Mesh convergence of $\langle E_{area} \rangle$ for algebraic formulae applied to f and ϕ*

In Fig. 8.3a, convergence of $\langle E_{area} \rangle$ is presented for the algebraic formulae applied to the volume fraction $H^\epsilon_\Gamma(f)$. The sharpest version AF2 provides highly erroneous area estimations while AF1 shows convergence in the low resolution. In the high resolution limit, AF0 and AF1 gives similar results. However, none of these formulae converges with mesh refinement. On the other hand, the GFM formulation presents a second-order convergence which is always more accurate than the algebraic formulae considered here.
For completeness, Fig. 8.3b shows $\langle E_{area} \rangle$ convergence of those formulae applied to $H^\epsilon_\Gamma(\phi)$ which is a smoother version of $H^\epsilon_\Gamma(f)$. The smoothing of H_Γ leads to higher errors on AF0, however it provides a convergent behaviour for AF1.
Overall, the conclusions are the same than in the literature: using algebraic formulae is not suitable for accurate and sharp evaluation of the interface area. In [202], a complete comparison of different interfacial area calculations was performed. It showed that the PLIC reconstruction always provide a better approximation than the algebraic formulas defined above. It was also observed that the spherical shape of an evaporating droplet was preserved using $\mathscr{F}(f) = 6f(1-f)$ but with an under-predicted volume change.
Finally, an important observation can be done for the GFM formulation. Surprisingly, using the sharpest Heaviside H^0_Γ leads to a second-order evaluation of the interface area. This explains why GFM applied to normal gradient jump converges. Indeed, a normal gradient jump traduces a flux jump which is purely related to surface exchanges. Then, if the area of this surface exchange is well reproduced, the method is then consistent and leads to accurate results.
Even if the GFM method gives an accurate representation of the interface, it only provides A_Γ while other geometric properties such as wetted areas A_f and phase fraction f are needed

for applying embedded boundary conditions with a cut-cell method (see Section 8.3.1). Then, explicit reconstructions are investigated which can provide all the interface topology quantities required.

8.1.2 PLIC reconstruction

In the geometric VOF framework, a PLIC reconstruction of the interface is naturally available and can be used directly to compute A_Γ, A_f and f.
It is also possible to construct a mean plane from the $\phi^{-1}(0)$ isocontour intersection with the mesh and perform a PLIC reconstruction from this plane as proposed in the open source solver Basilisk [171]. This method was originally used to initialise the VOF field from any Level Set function, but it can be adapted to interface reconstruction in a SLS framework. The algorithm for a 2D reconstruction is provided here for a given cell, knowing the cell-centered values of ϕ:

1. The ϕ values are computed at the corners from simple linear interpolation.
2. Check if it is a mixed cell:
   - The four corner values of ϕ have the same sign. Then the cell is pure and does not need reconstruction. The algorithm is over.
   - At least one ϕ value has a different sign (case of Fig. 8.4a where ϕ_1 and ϕ_2 are positive and ϕ_3 and ϕ_4 are negative). The cell is mixed and requires a PLIC reconstruction.
3. Find the intersection between the edges of the cell and the interface:
   - The two vertices have the same sign (e_{12} is in the liquid while e_{34} is in the gas in Fig. 8.4b). Then the edge is not cut by the interface.
   - The two vertices have different sign (case of e_{14} and e_{23} in Fig. 8.4b). Then the edge fraction f^e and position of the interface $\mathbf{x}_\Gamma$ are computed based on the ϕ values at the vertices. Based on the example of Fig. 8.4b, the fraction f^e_{23} for edge e_{23} is computed as

$$f^e_{23} = \frac{\phi_2}{\phi_2 - \phi_3} \quad . \tag{8.9}$$

4. Compute the average normal $\overline{\mathbf{n}}$ based on the edge fractions

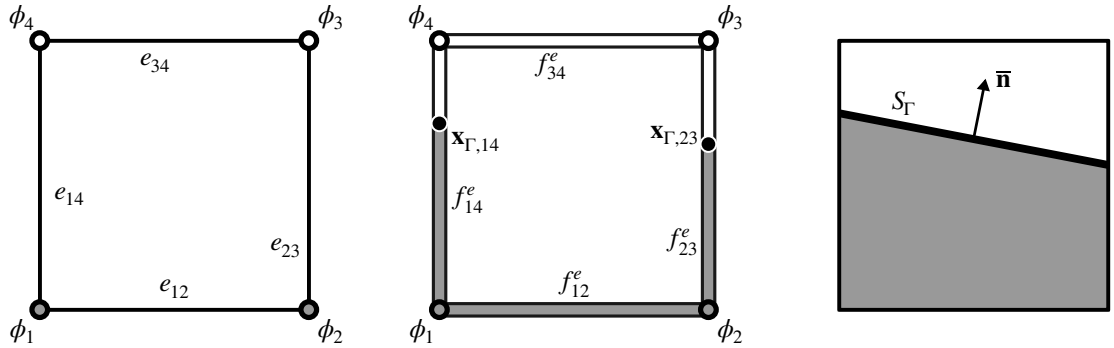
$$\overline{n}_x = \frac{f^e_{14} - f^e_{23}}{\Delta x} \quad , \tag{8.10}$$

$$\overline{n}_y = \frac{f^e_{12} - f^e_{34}}{\Delta y} \quad . \tag{8.11}$$

   This average normal is then normalized.
5. Compute the average plane parameter $\overline{d}$ based on interface positions $\mathbf{x}_\Gamma$, for edge e_{23}, this gives

$$d_{23} = \mathbf{x}_{\Gamma,23} \cdot \overline{\mathbf{n}} \quad . \tag{8.12}$$

   Then the average plane parameter $\overline{d}$ is retrieved by mean of all intersection evaluations of d.

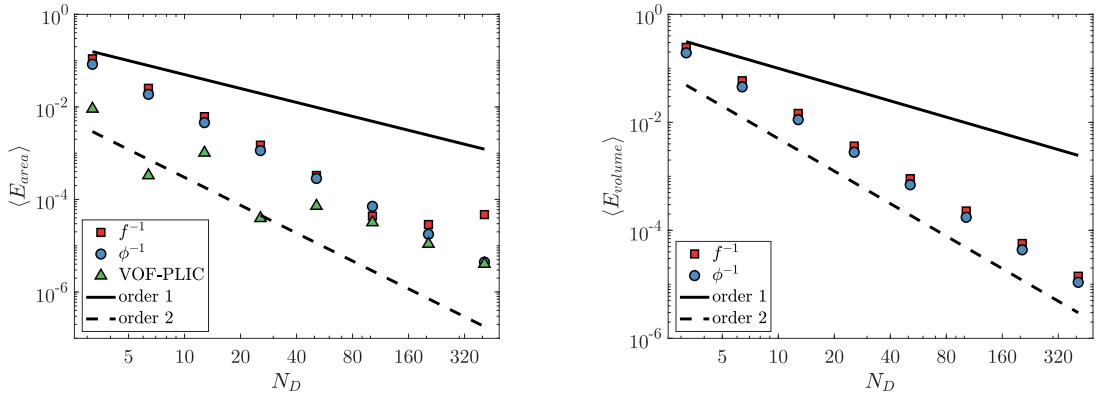


(a) *Step 1* **(b)** *Step 3* **(c)** *Final reconstruction*

Figure 8.4: *PLIC reconstruction from a Level Set field ϕ on a 2D Cartesian grid*

Once the PLIC reconstruction is available, it is simple to compute the surface A_Γ and volume fraction f on cartesian grids using analytic relations of [193]. Note that this algorithm can be applied to any color function by considering the appropriate isocontour c^{-1}.

The investigations then focus on the PLIC reconstruction of an interface either from the algorithm based on the isocontour c^{-1} or directly from the PLIC reconstruction obtained from the algorithm of Section 3.1.1. While the approach from an isocontour is not exact in volume, the inherent VOF-PLIC reconstruction has a natural constraint on volume conservation and volume error $\langle E_{volume} \rangle$ is not represented.



(a) *Convergence of $\langle E_{area} \rangle$* **(b)** *Convergence of $\langle E_{volume} \rangle$*

Figure 8.5: *Mesh convergence of $\langle E_{area} \rangle$ and $\langle E_{volume} \rangle$ for PLIC reconstruction from f^{-1}, ϕ^{-1} and from the classic VOF-PLIC procedure*

In Fig. 8.5a, the convergence of $\langle E_{area} \rangle$ is presented where the VOF-PLIC reconstruction shows higher accuracy for all resolutions while the f^{-1} contour-based reconstruction does not converge for the highest resolutions. Then, for VOF method, it is still better to rely on the inherent PLIC reconstruction as it is already available after the transport step

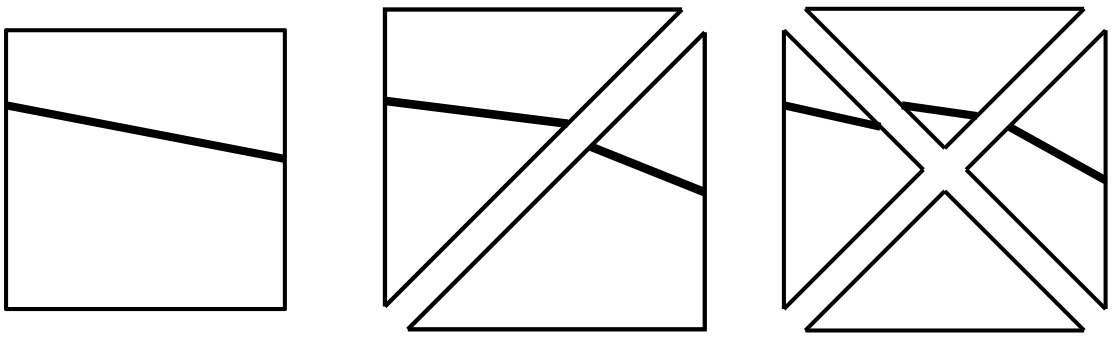
and remains more accurate. Moreover, Fig. 8.5b shows second-order accuracy of the f^{-1} contour-based reconstruction for volume evaluation while the inherent PLIC reconstruction is exact.
The method converges to second-order for both $\langle E_{area} \rangle$ and $\langle E_{volume} \rangle$ in the SLS framework.
Note that the final PLIC segment does not have to match exactly with the intersection points. Some authors [5; 102] argued that the discontinuous nature of the PLIC reconstruction could lead to error in the interface surface calculation even if it provides an exact volume. Moreover, the discontinuous nature of such reconstruction leads to different wetted areas A_f at the same cell face, which can be problematic for cut-cell methods (see Section 8.3.3.3 for a discussion).
To address this issue, the isocontour c^{-1} can be used to construct another interface which is connected to the neighbouring pieces. Such geometric reconstruction based on the isocontour can be done with a cell decomposition method.

8.1.3 Cell decomposition

The different cell decompositions are defined as follows

- CD0 is the marching cube method
- CD1 is the simplex decomposition in two simplices
- CD2 is the simplex decomposition in four simplices

CD0 corresponds to a zero-level decomposition where the intersections are computed in the computational cell. CD1 corresponds to a one-level decomposition where the computational cell is decomposed into two simplices. CD2 corresponds to a two-level decomposition where the two simplices obtain from CD1 are again decomposed into to simplices.



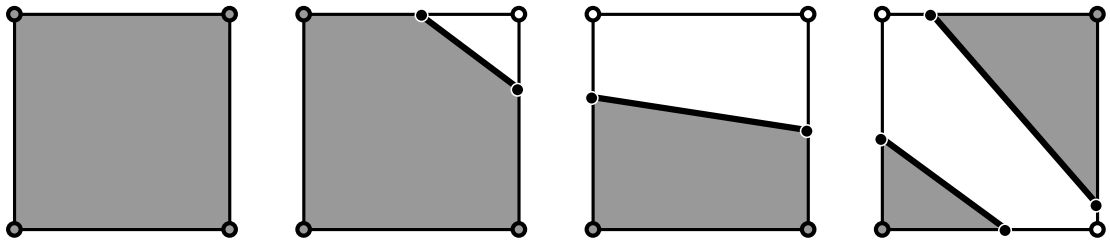
(a) *CD0* **(b)** *CD1* **(c)** *CD2*

Figure 8.6: *Cell decomposition level illustration, the thick line represent a piece of interface*

The difference between these three levels of decomposition is illustrated in Fig. 8.6. While CD0 leads to a single linear reconstruction in the cell (Fig. 8.6a), CD1 allows two slopes (Fig. 8.6b) and CD2 three slopes (Fig. 8.6c). Then, the level of decomposition can provide a subcell representation of the interface curvature if a simplex decomposition is used. Now that the cell decomposition concept is introduced, the different algorithms are detailed.

8.1.3.1 Marching cube

From medical data processing [120], the Marching Cube (MC) uses a domain decomposition into cubes with intersections to an isosurface. As for the previous method, the idea is to find the intersections between the Cartesian mesh and the isosurface. Each vertex value ϕ is either negative or positive which leads to $2^4 = 16$ different triangulation in 2D or $2^8 = 256$ in 3D.



(a) *Case 1* **(b)** *Case 2* **(c)** *Case 3* **(d)** *Case 4*

Figure 8.7: *Marching Cubes configurations on a 2D Cartesian*

However, by using primary geometric transformations such as rotation and symmetry, the number of triangulations is only of 4 in 2D (see Fig. 8.7) and 15 in 3D. It is then straightforward to retrieve the surface and volume of the triangulation from a given configuration. The algorithm of MC is the following:

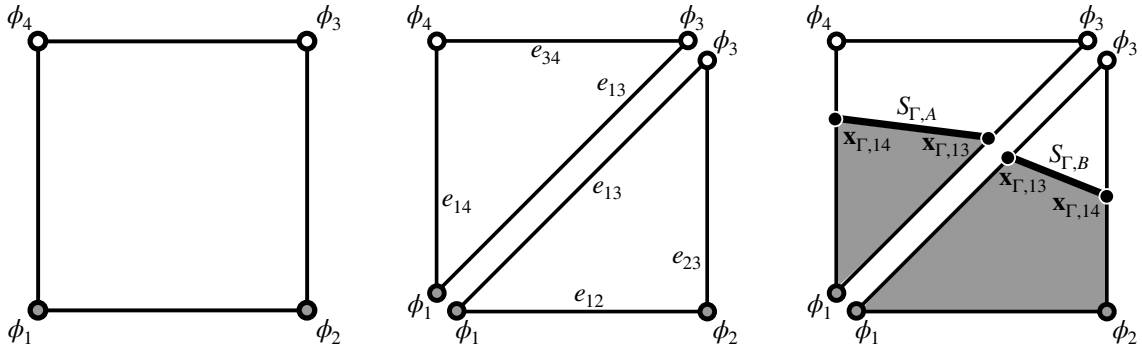
1. The ϕ values are computed at the corners from simple linear interpolation
2. Perform rotation and symmetry in order to have all normal component positives with $n_x < n_y$
3. Check the four corner values of ϕ and deduce the MC case
4. From the MC case and intersection points, compute the surface and volume of the reconstruction

The example presented in Fig. 8.4 corresponds to the case 3 in Fig. 8.7c where two corners are in liquid and two are in gas.
In [24], a Marching Cube algorithm applied to the $f^{-1}(0.5)$ isocontour shows improvements in the area evaluation compared to a PLIC reconstruction. The same study is preformed in the following. Note that PLIC and MC methods are limited to Cartesian grids.
For more general meshes, another decomposition can be performed using simplex decomposition (SD).

8.1.3.2 Simplex decomposition

The simplex decomposition method, due to [134], decomposes the computational cell into simplices. Then, the interface can be reconstructed from the intersections between $\phi^{-1}(0)$ isocontour and the simplices. The choice of using the simplex as a primitive shape is attractive for its simple surface and volume evaluations.
The minimum number of simplices to build a square is 2, and this is the choice made in the original decomposition as in Fig. 8.8. In fact, this decomposition allows to have two slopes

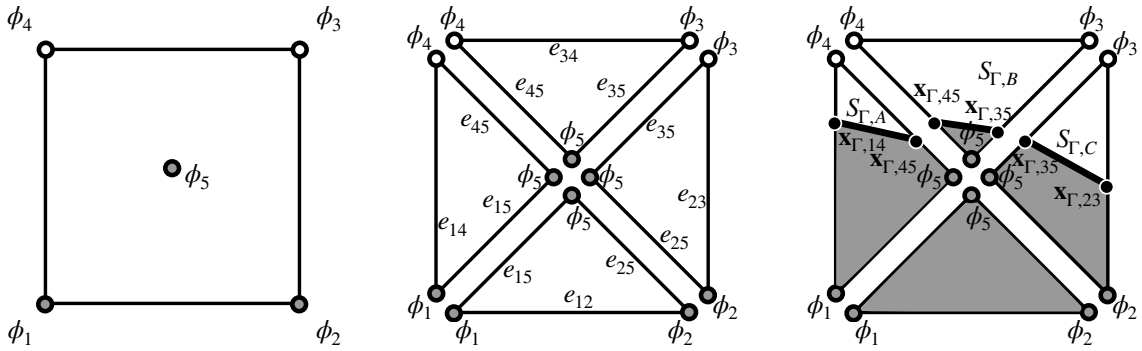
in a single computational cell which is not captured in the case of a PLIC reconstruction and MC method.



(a) *Step 1* **(b)** *Step 2* **(c)** *Final decomposition*

Figure 8.8: *Simplex decomposition from a Level Set field ϕ on a 2D Cartesian grid using 2 simplices*

While this algorithm is only based on the corner values of ϕ, Alis [4] proposed to add the center value of ϕ which allows a cell decomposition into four simplices instead of two (see Fig. 8.9). This new decomposition allows three different slopes in a single computational cell which is expected to be more accurate for curved interfaces.



(a) *Step 1* **(b)** *Step 3* **(c)** *Final reconstruction*

Figure 8.9: *Simplex decomposition from a Level Set field ϕ on a 2D Cartesian grid using four simplices*

The algorithm for a 2D decomposition is provided here (3D version follows the same philosophy).

1. The ϕ values are computed at the corners from simple linear interpolation
2. Cut the cell into two (or four) simplices
3. For each simplex, find the intersection between the edges and the interface using the vertex values of ϕ

Then, the evaluation of geometric properties is straightforward for a simplex. The total interface area of a cell is then the sum of all interface areas captured by the simplices. For

the example given in Fig. 8.8c, the surface area is $A_\Gamma = S_{\Gamma,A} + S_{\Gamma,B}$ while for Fig. 8.9c the surface area is $A_\Gamma = S_{\Gamma,A} + S_{\Gamma,B} + S_{\Gamma,C}$.

8.1.3.3 Comparison of different level of decomposition

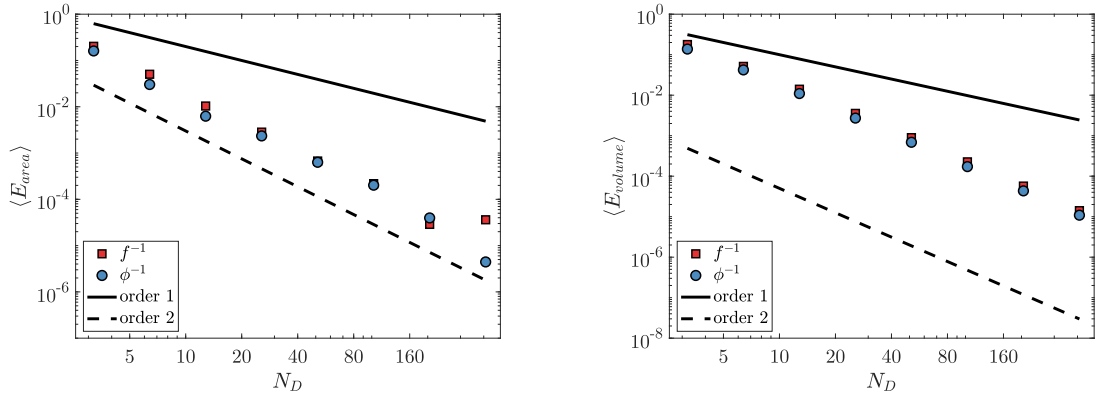
As the above methods involve increasing implementation and computational effort with the increasing level of decomposition, it is interesting to investigate the associated gain in accuracy. Moreover, these decompositions can be applied to either f^{-1} or ϕ^{-1} which are of different nature and are not expected to provide the same results.
The area and volume errors from these three decomposition levels is presented in Fig. 8.10. In Fig. 8.10a, a second-order convergence of the area evaluation is observed for the use of CD0 with small effects of the choice of c^{-1} on the results. However, f^{-1} seems to saturate for the last resolution point displayed. This saturation is not observed for $\langle E_{volume} \rangle$ in Fig. 8.10b where a second-order convergence of the volume is also obtained. The saturation of $\langle E_{area} \rangle$ for evaluation from f^{-1} is happening for coarser meshes using CD1 and even coarser meshes using CD2 as displayed in Figs. 8.10c and 8.10e. However, this saturation is still not observed for the convergence of $\langle E_{volume} \rangle$ in Figs. 8.10d and 8.10f. On the other hand, all the decompositions applied to ϕ^{-1} show a second-order convergence with accuracy magnitude increased by the level of decomposition.
From this observations, a conclusion can be drawn: the evaluation of area seems to be more prompt to errors compared to the volume and increasing the subcell degree of freedom amplifies these errors for f^{-1}. This can be explained by the following considerations:

- The area is indirectly related to the first-order derivatives of c while the volume depends directly on c itself.
- Linear interpolations are used to find the intersections in the geometric reconstruction process.
- Increasing the level of decomposition also increases the number of intersections needed for the subcell linear reconstruction.

In practice, f is far from being linear as it is a discontinuous function with an intermediate region width dependant of the mesh size. Then, the intersections are prompt to errors which are accumulated by increasing the level of decomposition. These errors are not amplified when computing the volume as it is related to c itself instead of the first derivative.

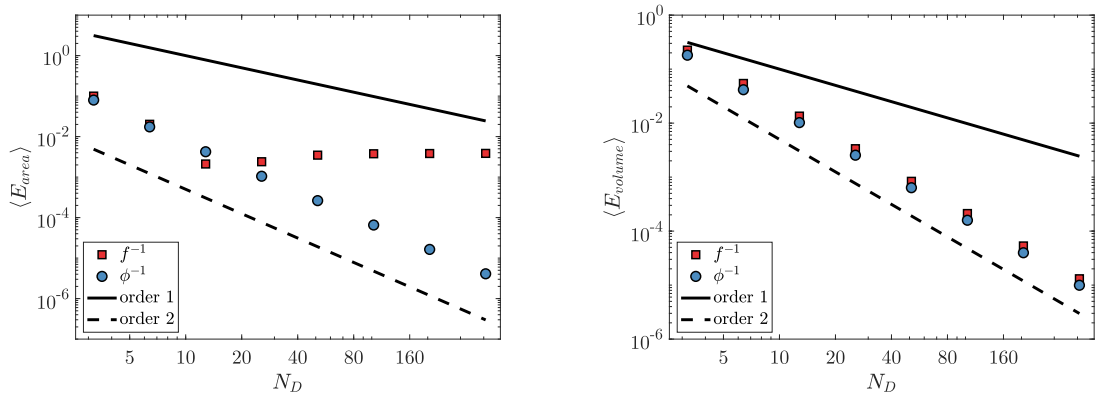
8.1.4 Method choice recap for interface reconstruction

After the above investigations, it is possible to choose the most accurate reconstruction methodology for VOF and SLS. To compare the most promising methodologies applied to a given capturing method, Fig. 8.11a compares GFM, PLIC-VOF and CD0 applied to VOF while Fig. 8.11b compares GFM, PLIC from ϕ^{-1} and CD2 applied to SLS.
For VOF, using the inherent PLIC reconstruction leads to the most accurate area evaluation and does not require additional reconstruction. Then, in the solver, embedded boundaries will rely on the PLIC reconstruction and the collocated Dirac appearing in the continuity equation will be computed as $\delta_\Gamma = A_\Gamma / V_\mathcal{C}$.
For SLS, the CD2 and GFM methods provide equivalent surface approximations. Then



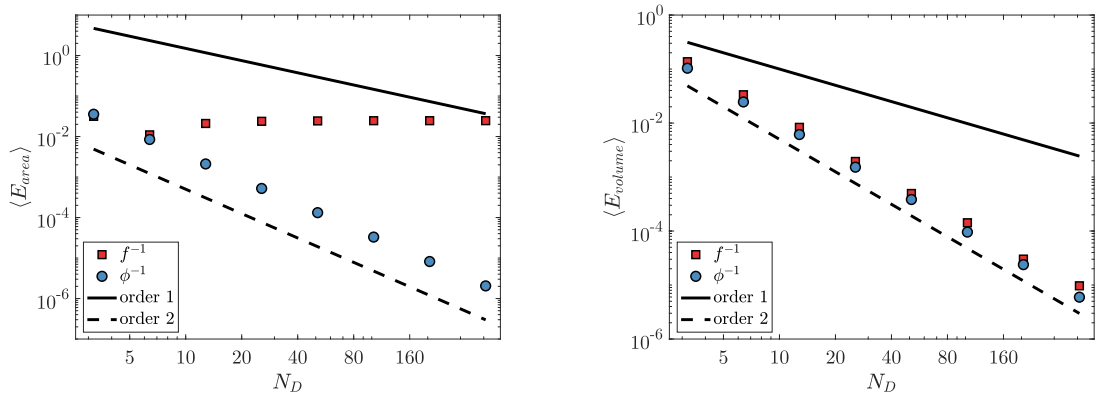
(a) *Convergence of* $\langle E_{area} \rangle$ *for CD0*

(b) *Convergence of* $\langle E_{volume} \rangle$ *for CD0*



(c) *Convergence of* $\langle E_{area} \rangle$ *for CD1*

(d) *Convergence of* $\langle E_{volume} \rangle$ *for CD1*

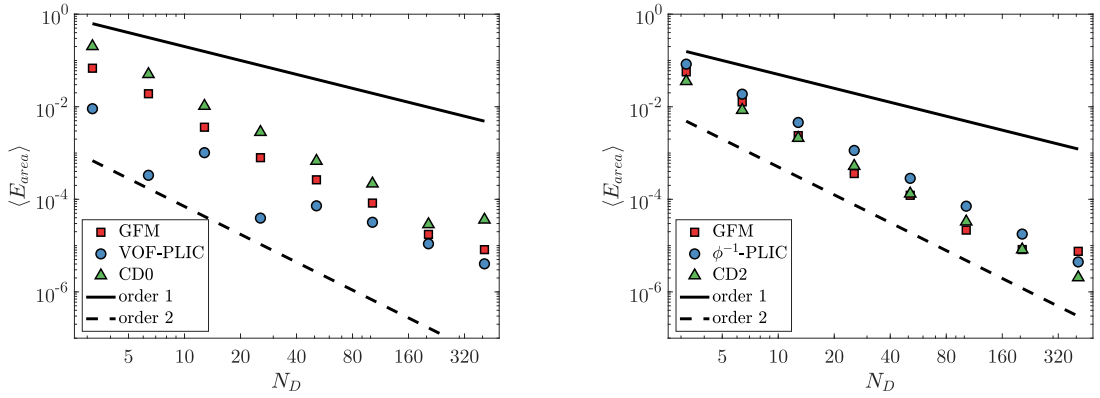


(e) *Convergence of* $\langle E_{area} \rangle$ *for CD2*

(f) *Convergence of* $\langle E_{volume} \rangle$ *for CD2*

Figure 8.10: *Mesh convergence of* $\langle E_{area} \rangle$ *and* $\langle E_{volume} \rangle$ *for the different cell decomposition levels CD0, CD1 and CD2*

if a cut-cell method is used for embedded boundaries, the CD2 is used to reconstruct the interface because GFM only provides the area while CD2 also provides all the other



(a) *VOF* **(b)** *SLS*

Figure 8.11: *Mesh convergence of $\langle E_{area} \rangle$ for different reconstruction methods applied to VOF and SLS*

geometric properties required for the cut-cell method. However, if a ghost-cell method is used for embedded boundaries, the GFM approximation is used to avoid the expensive simplex decomposition. The summary of choices for δ_Γ and A_Γ

method	VOF	SLS
δ_Γ	PLIC	GFM
A_Γ, A_f and f	PLIC	CD2

Table 8.1: *Computation choices for VOF and SLS reconstruction of δ_Γ, A_Γ, A_f and f*

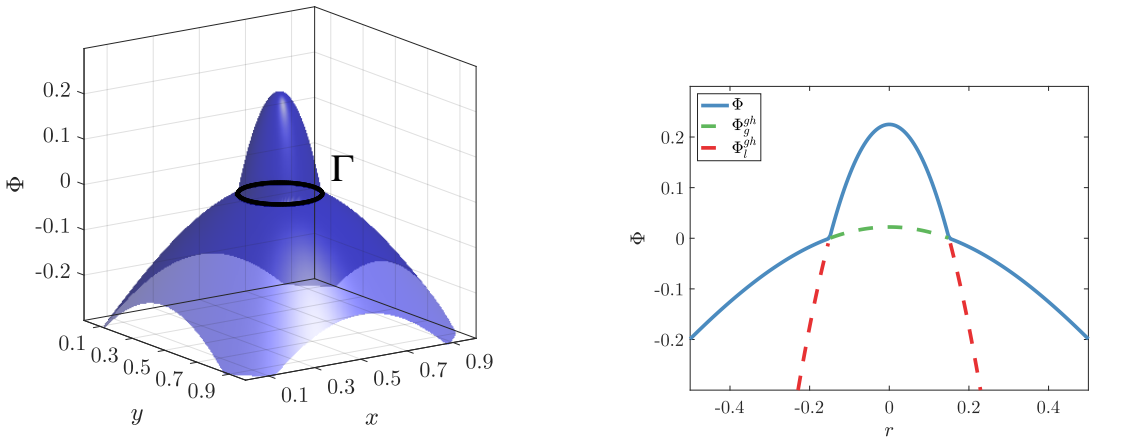
8.2 Quantities at the interface

An accurate approximation of the normal gradient at the interface is essential for phase-change simulations as it is used for computation of $\dot{m}$. This numerical aspect has been broadly investigated in the literature for phase-change simulations as it will directly impact the accuracy of the predicted mass variation during a simulation. The main challenge of this computation is to provide high-order accuracy without using averaged quantities close to the interface. Several approaches can respond to these constraints such as one-sided differences, ghost cell differences or functional fitting. In this section, these methods are evaluated and improvements are proposed to obtain an accurate gradient computation for VOF and SLS representation of the scalars.
Moreover, T_Γ and Y_Γ have to be reconstructed at the interface for Dirichlet boundary condition imposition. The ghost cell method and the functional fitting also have the capability to provide the value of a quantity at the interface. The study is then extended to such reconstruction.
For a quantitative investigation of the gradient reconstruction accuracy, a scalar Φ is defined as in [7] with associated illustration in Fig. 8.12 in a domain $\Omega = \Omega_l \cap \Omega_g$ of dimensions

$[1 \times 1]$ where Ω_l is the interior and Ω_g the exterior of the circle of radius $R = 0.2$ centered at $(0.5, 0.5)$

$$\Phi = \begin{cases} \exp\left(-r^2\right) \exp - \left(-R^2\right) & \text{if } r > R \\ \Phi = 10\left(R^2 - r^2\right) & \text{if } r \leq R \end{cases} \quad , \tag{8.13}$$

with r the reduced radius such that $r = \sqrt{(x-0.5)^2 + (y-0.5)^2}$.



(a) *Surface contour of Φ with the interface Γ represented in black.*

(b) *Evolution of Φ with respect to r (in blue) with ghost values (in red and green).*

Figure 8.12: *Illustration of Φ with a surface contour (left) and a 2D plot with respect to r (right)*

Then the value of Φ at the interface is $\Phi_\Gamma = 0$ and the normal gradients from the liquid and gas phases are

$$\Phi^{(1)}_{\Gamma,l} = -20R \quad , \tag{8.14}$$

$$\Phi^{(1)}_{\Gamma,g} = -2R\exp\left(-R^2\right) \quad . \tag{8.15}$$

The errors for a phase normal gradient $\Phi^{(1)}_{\Gamma,p}$ with $p = l, g$ at the interface is computed as

$$L_2\left(\Phi^{(1)}_{\Gamma,p}\right) = \frac{\sqrt{\frac{1}{N_\Gamma}\sum_{i=1}^{N_\Gamma}\left(\Phi^{(1)}_{\Gamma,p,exact} - \Phi^{(1)}_{\Gamma,p,i}\right)^2}}{\Phi^{(1)}_{\Gamma,p,exact}} \quad . \tag{8.16}$$

Note that it is interesting to investigate both gas and liquid normal gradients as there related fields are not of the same nature.

8.2.1 One-sided differences

The idea behind one-sided differences is to include the interface value Φ_Γ in the stencil and only consider values defined in a single phase. This implies the use of the interface location

$\mathbf{x}_\Gamma$ from either the PLIC reconstruction or the signed distance with the methods detailed in Section 8.2.1.1. It is used as a direct method to approximate the normal gradient at the interface. Note that in such case, the interface value Φ_Γ needs to be deduced from one of the methods presented below. However, this method can also be used to close the system presented in Section 7.1. In such case, Φ_Γ and its normal derivative $\Phi_\Gamma^{(1)}$ are obtained through an iterative process.

Directional-splitting First, one-sided gradients can be evaluated with directional-splitting from $\Phi_\Gamma^{(1)} = \nabla\Phi \cdot \mathbf{n}_\Gamma$. This has been proposed in the VOF framework [195] by using the PLIC reconstruction for the interface location $\mathbf{x}_\Gamma$. In a 2D cell $\mathcal{C}_i$, the first component $\frac{\partial\Phi}{\partial x}\big|_{\Gamma,i}$ is computed as

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i} = \frac{\Phi_{i+1} - \Phi_{\Gamma,i}}{\Delta x + \Delta x_\Gamma} \quad , \tag{8.17}$$

with Δx_Γ the distance between x_i and x_Γ. In this expression, the computational point is always chosen such that $\Delta x + \Delta x_\Gamma > \Delta x$ to avoid any troublesome behavior when $\mathbf{x}_\mathcal{C}$ is too close to $\mathbf{x}_\Gamma$. In the Level Set framework, the same type of one-sided differences can be designed using the signed distance ϕ. A first discretization was presented by [205] where, in a cell $\mathcal{C}_i$ with the interface lying between x_i and x_{i+1}, the gradient is computed as

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i} = \frac{\Phi_{i+1} - \Phi_{\Gamma,i}}{\theta\Delta x} \quad , \tag{8.18}$$

with $\theta = |\phi_i|/(|\phi_i|+|\phi_{i+1}|)$. The finite differences of Eqs. (8.17) and (8.18) are second-order accurate but defined at $\mathbf{x}_\mathcal{C}$. Analogously to the curvature computation, an evaporation rate computed at cell centres contains an inherent first-order error as explained in Section 5.4.2.3, this point is detailed hereafter in Section 8.2.2.2.
In [190], a higher-order finite difference defined in cell $\mathcal{C}_i$ is used if the interface lies between x_i and x_{i-1}

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i} = \frac{-\Delta x_\Gamma^2\Phi_{i-1} + \left(\Delta x_\Gamma^2 - \Delta x^2\right)\Phi_i + \Delta x^2\Phi_{\Gamma,i}}{\Delta x\Delta x_\Gamma\left(\Delta x + \Delta x_\Gamma\right)} \quad . \tag{8.19}$$

In [7], an improvement is proposed to handle the singular case $\Delta x_\Gamma \approx 0$

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i} = \frac{\left(\Delta x^2 - \Delta x_\Gamma^2\right)\Phi_{i-2} + \left(-4\Delta x^2 + \Delta x_\Gamma^2\right)\Phi_{i-1} + 3\Delta x^2\Phi_{\Gamma,i}}{\Delta x\left(\Delta x + \Delta x_\Gamma\right)\left(2\Delta x + \Delta x_\Gamma\right)} \quad . \tag{8.20}$$

This new stencil is used only when $\Delta x_\Gamma < \epsilon_c$ defined as $\epsilon_c = 10^{-3}\Delta x$. Actually, it is possible to use any arbitrary order finite differences for this gradient computation. In [24] fourth-order one-sided gradients are used.
As those expressions corresponds to Taylor expansions of $\frac{\partial\Phi}{\partial x}\big|_{\Gamma,i}$ around x_Γ, the gradient is then defined naturally at the interface location.

Normal direction Other methods use an inherent normal-oriented gradient instead of using a dimensional-splitting. With the TFF of [159], the authors make use of the location of Φ at the phase barycenter $\mathbf{x}_{cm,i}$ to compute a compact normal gradient as

$$\Phi^{(1)}_{\Gamma,i} = \frac{\Phi_i - \Phi_{\Gamma,i}}{\|\mathbf{x}_{cm,i} - \mathbf{x}_\Gamma\|_2} \quad . \tag{8.21}$$

This expression leads to a first-order approximation of the interface not well-suited for accurate phase-change procedure and was only used for discretization of interface fluxes using cut-cell methods presented in Section 8.3.1. Moreover, $\|\mathbf{x}_{cm,i} - \mathbf{x}_\Gamma\|_2$ can be arbitrary close to zero.
In the Front-Tracking framework, a method based on a fixed normal distance to the interface is presented in [85] to prevent any singular behaviour for gradient discretizations. This takes the form of a first-order one-sided difference

$$\Phi^{(1)}_{\Gamma,i} = \frac{\widetilde{\Phi}(\delta) - \Phi_{\Gamma,i}}{\delta} \quad , \tag{8.22}$$

with $\widetilde{\Phi}(\delta)$ retrieved from linear interpolation of the cells adjacent to $\mathcal{C}_i$ at the point which is located at a normal distance δ from the interface. δ can be chosen between Δx and $2\Delta x$ without noticeable impact on the accuracy. It has also been used in a VOF framework relying on the PLIC reconstruction barycenter [236]. In [51], a second-order discretization is proposed

$$\Phi^{(1)}_{\Gamma,i} = \frac{\widetilde{\Phi}(2\delta) - 4\widetilde{\Phi}(\delta) + 3\Phi_{\Gamma,i}}{2\delta} \quad , \tag{8.23}$$

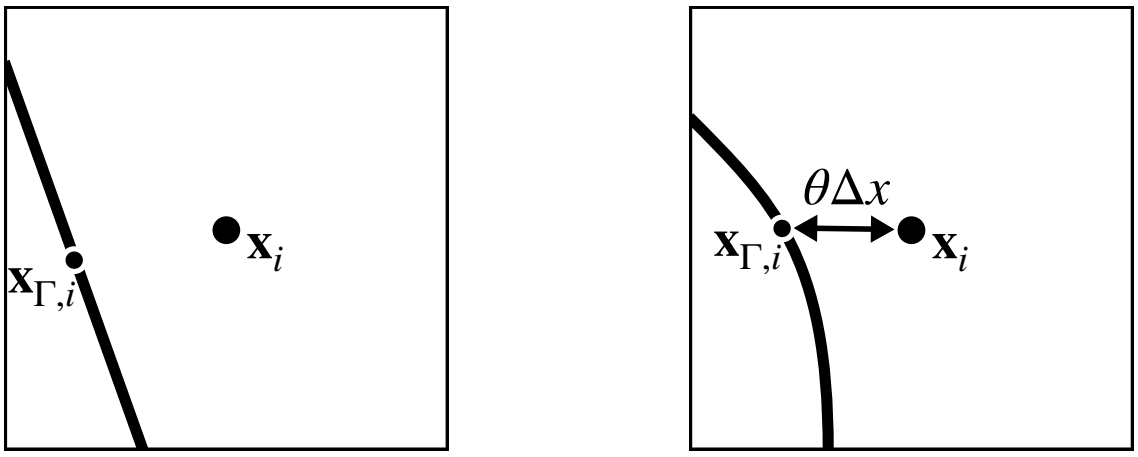
with $\widetilde{\Phi}(2\delta)$ and $\widetilde{\Phi}(\delta)$ obtained from linear interpolation.
A last approach was presented in a WDF [102; 72; 123] to avoid using the mixed cells quantities which are averaged. The normal gradient is computed in the set of cells Ω^1_p (or Ω^2_p in [123]) by first-order derivatives using Eq. (8.22) with $\widetilde{\Phi}(\delta) = \Phi_i$ for a given cell $\mathcal{C}_i$. Ω^1_p is the set of points in the first narrow band of pure cell in phase p while Ω^2_p includes first and second narrow bands. The mixed cell used for the computation is chosen in the mixed neighbour cells of $\mathcal{C}_i$ which maximizes collinearity defined as

$$\xi = \mathbf{n}_\Gamma \cdot \Delta\mathbf{x} \quad , \tag{8.24}$$

with $\mathbf{n}_\Gamma$ the normal of the PLIC reconstruction in the mixed cell and $\mathbf{x}$ the displacement vector between the considered cell center $\mathbf{x}_i$ and the cell center of the mixed cell. Then, the gradients in the mixed cells are retrieved either using simple average of neighbour pure cells [102; 72] or using a weighting [123]. The normal gradient is defined as

$$\Phi^{(1)}_{\Gamma,i} = \sum_{p \in \Omega^2_p} W_p \frac{T_p - T_\Gamma}{\delta_p} \quad , \tag{8.25}$$

with the weight W_p a function of the collinearity ξ of the neighbours.



(a) *VOF* **(b)** *SLS*

Figure 8.13: *Definition of* $\mathbf{x}_{\Gamma,i}$ *for VOF and SLS representations*

8.2.1.1 Interface location

In a one-sided discretization, the interface location is explicitly used in the stencil.
It is then important to evaluate the interface location error associated to a PLIC representation or to the signed distance ϕ:

- For a VOF representation, the interface location in a given cell $\mathbf{x}_{\Gamma,i}$ is defined by the interface fragment barycenter as illustrated in Fig. 8.13a.
- For a SLS representation, the interface location is defined between two cell centres by linear interpolation. If the interface lies between $\mathbf{x}_{i-1}$ and $\mathbf{x}_i$ as in Fig. 8.13b then

$$\mathbf{x}_{\Gamma,i} = \theta \mathbf{x}_{i-1} + (1-\theta)\mathbf{x}_i \quad , \tag{8.26}$$

  with $\theta = |\phi_i|/(|\phi_{i-1}| + |\phi_i|)$.

The error on the interface location is defined as in [7] : the exact signed distance at the interface is zero by definition while the computed signed distance based on the numerical interface location is

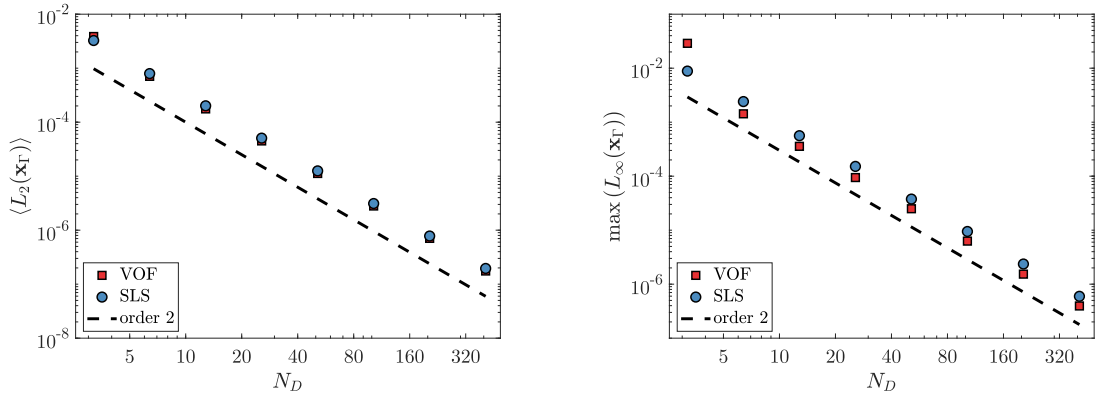
$$\phi\left(\mathbf{x}_{\Gamma,i}\right) = R - r(\mathbf{x}_{\Gamma,i}) \quad , \tag{8.27}$$

with r defined as in Section 8.3.3. Then, the error is computed as

$$L_2(\mathbf{x}_\Gamma) = \sqrt{\frac{1}{N_\Gamma}\sum_{i=1}^{N_\Gamma} \phi\left(\mathbf{x}_{\Gamma,i}\right)^2} \quad , \tag{8.28}$$

$$L_\infty(\mathbf{x}_\Gamma) = \max_i \phi\left(\mathbf{x}_{\Gamma,i}\right) \quad . \tag{8.29}$$

The errors are evaluated on 100 circles randomly located in the domain to meet as much configurations as possible. The final metric is $\langle L_2(\mathbf{x}_\Gamma)\rangle$ the mean of $L_2(\mathbf{x}_\Gamma)$ and $\max\left(L_\infty(\mathbf{x}_\Gamma)\right)$ the maximum of $L_\infty(\mathbf{x}_\Gamma)$ over all those configurations.
Both representations show a second-order accuracy in Fig. 8.14 which could lead to errors on one-sided gradient as demonstrated hereafter.



(a) *Convergence of* $\langle L_2(\mathbf{x}_\Gamma)\rangle$ **(b)** *Convergence of* $\max(L_\infty(\mathbf{x}_\Gamma))$

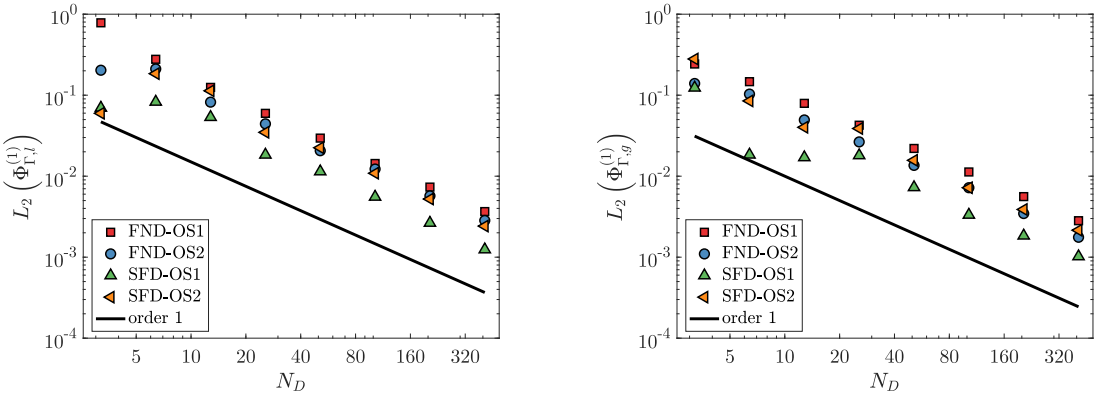
Figure 8.14: *Mesh convergence of* $\langle L_2(\mathbf{x}_\Gamma)\rangle$ *and* $\max(L_\infty(\mathbf{x}_\Gamma))$ *based on VOF or SLS*

8.2.1.2 Comparison of one-sided discretizations

In this comparison, four methodologies are presented, two based on the TFF and two based on the JCF:

- Fixed normal distance with one probe point Eq. (8.22) (FND-OS1)
- Fixed normal distance with two probe points Eq. (8.23) (FND-OS2)
- Dimensional-splitting finite differences using a compact stencil Eq. (8.18) (SFD-OS1)
- Dimensional-splitting finite differences using an extended stencil Eq. (8.20) (SFD-OS2)

For VOF, a natural discretization based on one-sided gradient makes use of the PLIC barycenter to build a gradient stencil in the normal direction. The most suited method using these information is the fixed distance FND-OS1 and FND-OS2.
For SLS, it is more natural to use the dimensional-splitting formulation of the gradient as a position in the normal direction of the interface is not straightforward. Then, SFD-OS1 and SFD-OS2 are used.
In Fig. 8.15, all methods exhibit a first-order convergence on the gradient. This rate is expected for FND-OS1 and SFD-OS1 while it is not trivial for FND-OS2 and SFD-OS2. In fact, using the location of the interface in the stencil, which has been demonstrated to be second-order accurate prevents any one-sided discretization to reach high-order accuracy. This point was reported in [7] where a higher-order representation of the interface using Gradient Augmented Level Set allowed to recover a second-order accuracy of the gradients. Moreover, the interface value Φ_Γ is not known a priori for general phase-change simulations. The quantity is reconstructed with a limited order of accuracy which could induce even more error in the gradient computation. This aspect is not discussed in most of the work using the one-sided gradient method as it is mainly used in boiling applications where T_Γ is imposed at the saturation temperature of the pure liquid.
These limitations can be circumvented by keeping a uniform discretization based on a ghost cell method. Then, neither the interface position nor the interface value are required.



(a) *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ **(b)** *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$

Figure 8.15: *Mesh convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ *and* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$ *for the different one-sided gradient discretizations*

8.2.2 Ghost cell method

The ghost cell method (GCM) can be applied to compute values and gradients using a uniform discretization. In [218; 29; 192], ghost cells are defined in the unknown part of the domain, and linear interpolation and central differences are used to retrieve the value and the gradient respectively. This gives the second-order interpolation of Φ_Γ at $\mathbf{x}_\Gamma$

$$\Phi_\Gamma = (1-\theta)\Phi_{p,i} + \theta\Phi^{gh}_{p,i+1} \quad , \tag{8.30}$$

with $\theta = |\phi_i|/(|\phi_i| + |\phi_{i+1}|)$.
In 2D or 3D configuration, a dimensional-splitting approach is used where Φ^x_Γ, Φ^y_Γ (and Φ^z_Γ) are obtained from Eq. (8.30) and projected to the correct value Φ_Γ with

$$\Phi_\Gamma = \Phi^x_\Gamma n_x^2 + \Phi^y_\Gamma n_y^2 \quad . \tag{8.31}$$

While liquid values are often used in the literature, it can also be done using the gas values instead. By using the two approximations of Φ_Γ from liquid and gas phases, an average could be used to mitigate errors in the extrapolation.
The second-order difference located at the cell center is written as

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i} = \frac{\Phi^{gh}_{i+1} - \Phi_{i-1}}{2\Delta x} \quad . \tag{8.32}$$

Note that in this expression, $\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i}$ is not defined at the interface but at $\mathbf{x}_i$. In [198], the gradient are upwind to give a gradient evaluation closer to the interface

$$\left.\frac{\partial\Phi}{\partial x}\right|_{\Gamma,i} = \frac{\Phi^{gh}_{i+1} - \Phi_i}{\Delta x} \quad . \tag{8.33}$$

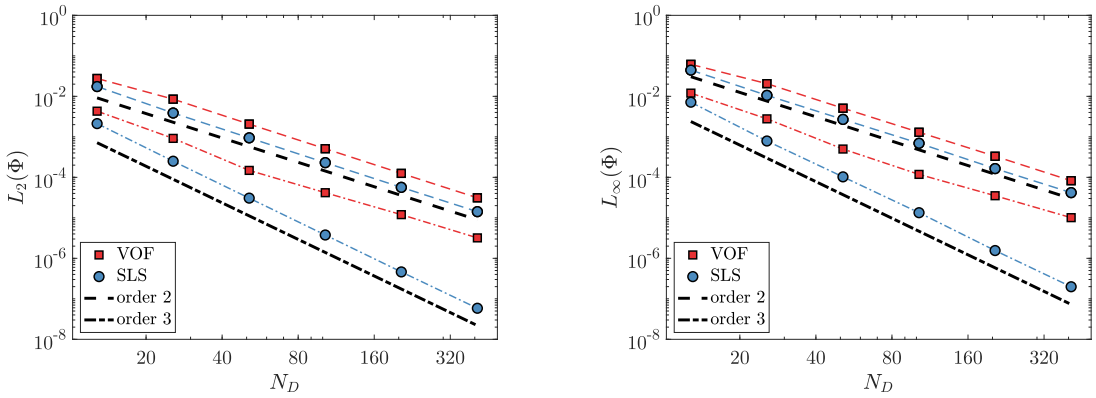
Even if this is still not providing $\nabla\Phi$ exactly at the interface position, it reduces the first-order error of a standard central difference located at the cell center by centering the gradient at the closest cell face to the interface.
The gradient is also treated in a dimensional-splitting fashion for multidimensional cases using $\Phi^{(1)} = \nabla\Phi \cdot \mathbf{n}$.

8.2.2.1 Extrapolation method

The main feature required for applying GCM is the extrapolation of scalars from one phase to another. The method used here is the PDE extrapolation presented in Appendix B.1, all the details in the implementation are provided in Appendix B.2. Here, only linear extrapolation (LE) and quadratic extrapolation (QE) are investigated either using VOF or SLS. The extrapolation is performed from Ω_l to Ω_g delimited by the circle of radius $R = 0.2$ centered at $(0.5, 0.5)$ in a domain $[1 \times 1]$. The target scalar field Φ is defined as in [30] in the whole domain

$$\Phi = \exp\left(4\,(x - 0.5)\,(y - 0.5)\right) \quad . \tag{8.34}$$

The goal of this test case is to extrapolate the scalar in the unknown domain and compare its values to the target scalar field.



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

Figure 8.16: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for VOF and SLS using linear extrapolation* (- - -) *or quadratic extrapolations* $N_D = 16$ (-···-).

In Fig. 8.16 are presented $L_2(\Phi)$ and $L_\infty(\Phi)$ for linear and quadratic extrapolations. It is interesting to notice that the normal approximation has an impact on extrapolation accuracy. The VOF representation provide less accurate normals compared to SLS which impacts the accuracy of the extrapolated field in both $L_2(\Phi)$ and $L_\infty(\Phi)$ metrics. This discrepancy is increasing for quadratic extrapolation and can have impacts on the gradient evaluation.
The interested reader can refer to Appendix B.1 for a more complete study of extrapolation accuracy.

8.2.2.2 Gradient location

A first-order trend is presented in Fig. 8.17 for gradients evaluated from both linear and quadratic extrapolations.
This is an expected results for linear extrapolation as the extrapolated field is second-order and the gradient computed using centered finite differences leads to a first-order gradient approximation. On the other hand, the quadratic extrapolation improves accuracy while maintaining a first-order convergence rate. From the same reasoning as for the linear extrapolation, a second-order gradient would be expected from a centered finite difference on a third order field.
In fact, the first-order trend is due to the location of the gradient evaluation. Analogous to the curvature computation, the gradients computed from uniform discretization are defined at the cell center and need an additional interpolation procedure to be correctly defined at the interface location. If no interpolation is performed, an inherent first-order error is introduced in the gradient evaluation and improving the accuracy of the field cannot overcome this limitation.
For higher-order gradients, an interpolation is then required. Again, this can be done either by linear or harmonic interpolation using the expression of Eqs. (5.69) and (5.70) applied to $\Phi^{(1)}$.
In practice, the gradient is first computed at the cell center $\mathbf{x}_i$. Then, the interface gradient is only computed in cells where ϕ changes sign as (here for $\phi_{i-1}\phi_i < 0$)

$$\Phi^{(1)}_{\Gamma,i} = \theta\Phi^{(1)}_{i-1} + (1-\theta)\Phi^{(1)}_{i} \quad . \tag{8.35}$$

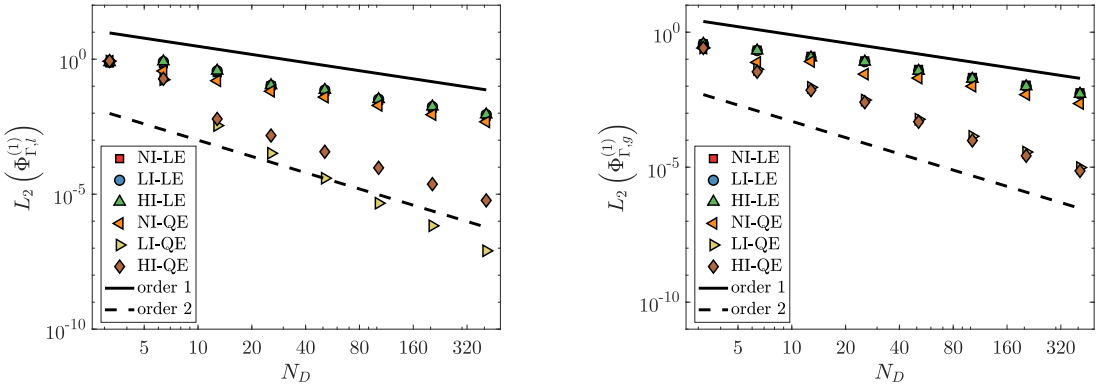
In a multi-dimensional case, ϕ can change sign in several directions, in this case, $\Phi^{(1)}_{\Gamma}$ is retrieved by weight average of all the interpolated values based on squared normal components. Note that this interpolation step is only required when using quadratic interpolation or higher-order. In the case of linear interpolation, the ghost cells are already defined such that the gradient is constant in the normal direction, then the evaluation of the gradient between two cells apart from an interface cells are expected to be substantially equal.

8.2.2.3 Comparison of the normal gradient reconstruction

For the comparison, six combination of field extrapolation and gradient interpolation are investigated in Fig. 8.17:

- No interpolation of the gradient with linear extrapolation of Φ (NI-LE)
- Linear interpolation of the gradient with linear extrapolation of Φ (LI-LE)
- Harmonic interpolation of the gradient with linear extrapolation of Φ (HI-LE)
- No interpolation of the gradient with quadratic extrapolation of Φ (NI-QE)
- Linear interpolation of the gradient with quadratic extrapolation of Φ (LI-QE)
- Harmonic interpolation of the gradient with quadratic extrapolation of Φ (HI-QE)

As expected, an interpolation process does not improve the accuracy of the interface gradient for linear extrapolations. Indeed, NI-LE, LI-LE and HI-LE collapse to the same error trend.
For quadratic extrapolation, the gradient is not constant anymore in the normal direction



(a) *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ **(b)** *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$

Figure 8.17: *Mesh convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ *and* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$ *for the different combination of field extrapolation and gradient interpolation*

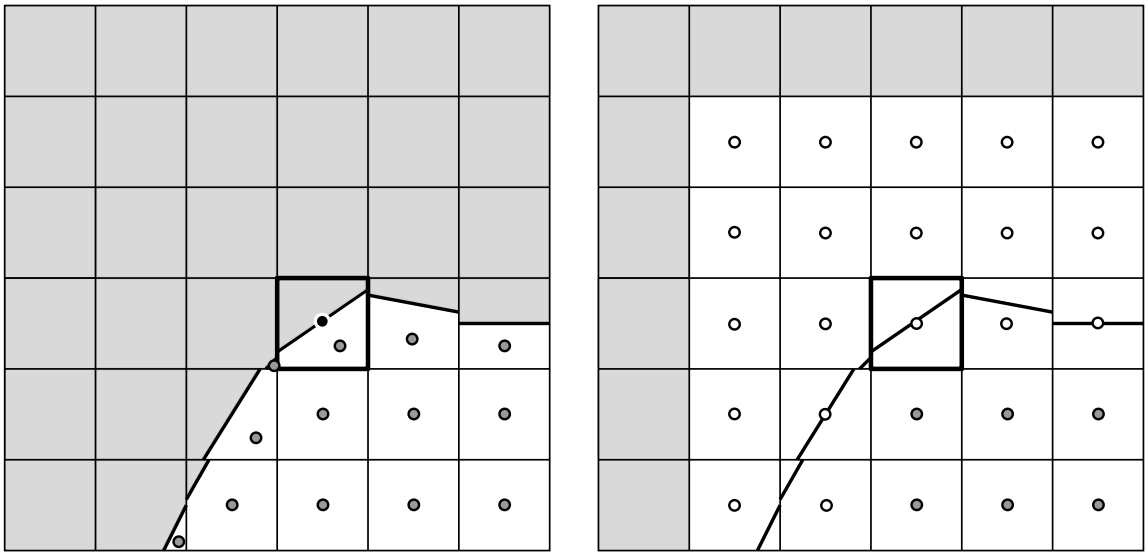
and an interpolation is expected to improve the accuracy of the evaluation. In Fig. 8.17, it is clear that both linear and harmonic interpolation improve drastically the accuracy and rate of convergence of the normal gradient evaluation. Surprisingly, a super-convergence is observed in Fig. 8.17a for the LI-QE with a normal gradient obtained between second and third order. This can be explained by the quadratic nature of the Φ_l field which lead to an exact extrapolation using the quadratic normal extrapolation, and an exact interpolation of the gradients using linear interpolation. For the exponential functional of Φ_g, the expected second-order convergence is retrieved in Fig. 8.17a.
As a conclusion, using a combination of quadratic extrapolation of Φ and linear or harmonic interpolation of $\Phi^{(1)}_\Gamma$ leads to second-order normal gradients.

8.2.3 Functional fitting

For the TFF, the methodology presented in Section 8.2.2 is not well-suited as no distance is available and the discretization close to the interface is inherently not uniform. This is why another method is proposed to increase the order of accuracy of the normal gradient by computing the value and normal gradient in mixed cells from a functional fitting based on least-square minimization of the field.
This idea has already been used in [159] to fit a linear profile for T and Y. As for the reconstruction of operators used in Section 5.4.2.1, this approach allows mitigation of numerical errors by increasing the sample size of points in the stencil. Moreover it can be applied to both uniform and non-uniform discretization without additional considerations.

8.2.3.1 Stencil definition

In the following, three different stencils are proposed to solve this problem either based on linear or quadratic fit. The different systems are illustrated in Fig. 8.18.



(a) *Phase stencil* **(b)** *Uniform stencil*

Figure 8.18: *Illustration of the set of points used to build the least square minimization in the mixed cell defined by the bold black line. The cells in grey are not included in the stencil. The points in grey are the transported values defined in the phase. The points in white represent the extrapolated values in the other phase. The black point is the interface value.*

Phase and interface stencil The first method proposes to only consider the points belonging to the considered phase with their values located at the phase barycenter (see Fig. 8.18a). Then a cell located in the phase p has the following distance to the phase barycenter $\Delta\mathbf{x} = \mathbf{x}_{cm,p} - \mathbf{x}_{\Gamma,i}$ and the value $\Phi(\mathbf{x}_{cm,p})$. An extension of this first system can be build by adding the interface value in the stencil. Then the associated distance is $\Delta\mathbf{x} = 0$ and the corresponding value is $\Phi(\mathbf{x}_{\Gamma,i}) = \Phi_\Gamma$.

Phase-restricted stencil It has been shown in Section 8.2.1 that errors on the interface location could lead to important errors in the gradient evaluation when the interface value was explicitly used in the stencil. To demonstrate this behaviour, another discretization is also proposed without including the interface value in the least square minimization. Moreover, the first stencil cannot be used for retrieving the interface value Φ_Γ as it is needed explicitly in the stencil.

Uniform stencil Finally, uniform discretization is also investigated. Then the distance is just $\Delta\mathbf{x} = \mathbf{x}_\mathcal{C} - \mathbf{x}_\Gamma$ with values taken at the cell center $\Phi(\mathbf{x}_\mathcal{C})$. When the stencil implies values of mixed cells or the other phase, the values are retrieved by normal extrapolation (white points in Fig. 8.18b). The values in the mixed also needs this extrapolation to maintain the uniform discretization.

Weighting A weight is inspired from the one-sided gradients Eq. (8.25) giving more importance to stencil points in the normal direction to the interface

$$W = \frac{\xi}{\|\Delta\mathbf{x}\|_2} \quad , \tag{8.36}$$

with ξ the collinearity defined by Eq. (8.24).

8.2.3.2 Functional definition

The least square minimization can be applied to different functional. In [159], the temperature is fitted on a linear profile while another approach proposed in this manuscript uses a quadratic fitting. The Taylor expansion around the interface location $\mathbf{x}_{\Gamma,i}$ gives

$$\Phi(\mathbf{x}) = \Phi(\mathbf{x}_{\Gamma,i}) + \Delta\mathbf{x}^\intercal\nabla\Phi(\mathbf{x}_{\mathcal{C}}) + \Delta\mathbf{x}^\intercal\nabla\nabla\Phi(\mathbf{x}_\Gamma)\Delta\mathbf{x} \quad , \tag{8.37}$$

with $\Delta\mathbf{x} = \mathbf{x} - \mathbf{x}_\Gamma$.

Linear fit If only the first-order derivatives are kept while second-order derivative are neglected, then the following linear system can be written

$$\begin{pmatrix} 1 & \Delta x_1 & \Delta y_1 \\ \vdots & \vdots & \vdots \\ 1 & \Delta x_N & \Delta y_N \end{pmatrix} \begin{pmatrix} \Phi(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial\Phi}{\partial x}(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial\Phi}{\partial y}(\mathbf{x}_{\Gamma,i}) \end{pmatrix} = \begin{pmatrix} \Phi(\mathbf{x}_1) \\ \vdots \\ \Phi(\mathbf{x}_N) \end{pmatrix} \quad . \tag{8.38}$$

This approach is expected to be first-order at the point of interest.

Quadratic fit Based on the complete Taylor expansion up to second-order derivatives, the following linear system is solved

$$\begin{pmatrix} 1 & \Delta x_1 & \Delta y_1 & \frac{1}{2}\Delta x_1^2 & \frac{1}{2}\Delta y_1^2 & \Delta x_1\Delta y_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 1 & \Delta x_N & \Delta y_N & \frac{1}{2}\Delta x_N^2 & \frac{1}{2}\Delta y_N^2 & \Delta x_N\Delta y_N \end{pmatrix} \begin{pmatrix} \Phi(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial\Phi}{\partial x}(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial\Phi}{\partial y}(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial^2\Phi}{\partial x^2}(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial^2\Phi}{\partial y^2}(\mathbf{x}_{\Gamma,i}) \\ \frac{\partial^2\Phi}{\partial x\partial y}(\mathbf{x}_{\Gamma,i}) \end{pmatrix} = \begin{pmatrix} \Phi(\mathbf{x}_1) \\ \vdots \\ \Phi(\mathbf{x}_N) \end{pmatrix} \quad . \tag{8.39}$$

This approach is expected to be second-order at the considered point of interest.

Erf function In [122], an Erf function is fitted at the interface vicinity following

$$\Phi(\widetilde{x}) = \begin{cases} C_1 + C_2\,\mathrm{erf}\,(C_3\widetilde{x}) & \text{if } \widetilde{x} \in \Omega_l \\ C_4 + C_5\,\mathrm{erf}\,(C_6\widetilde{x}) & \text{if } \widetilde{x} \in \Omega_g \end{cases} \quad , \tag{8.40}$$

with $\widetilde{x}$ the position coordinate in the normal direction to the interface such that the interface is located at $\widetilde{x}_\Gamma = 0$. The six coefficients are determined using the following information:

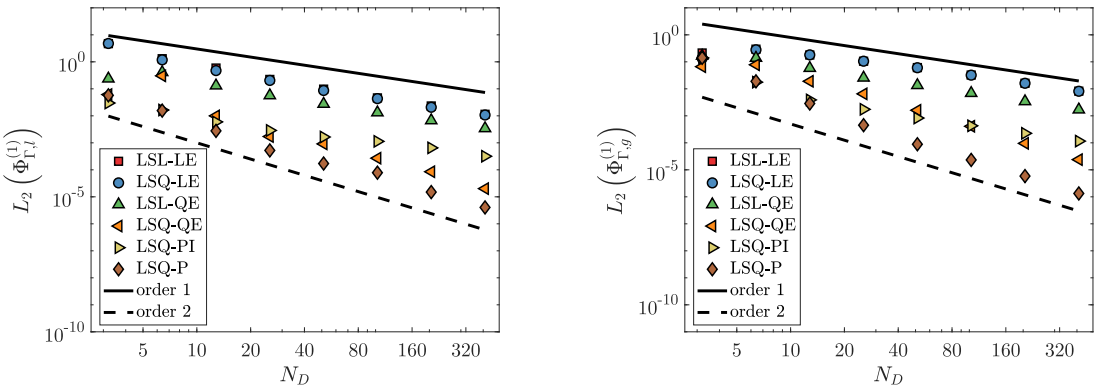
- Scalar evaluations $\Phi(-2\Delta x)$, $\Phi(-\Delta x)$, $\Phi(\Delta x)$, $\Phi(2\Delta x)$. Those points do not coincide with the cell center and are construction by linear interpolation.
- Continuity of Φ at the interface leading to $C_1 = C_4$.
- Closure for Φ_Γ (defined with Eq. (7.3) in the case of phase change with $\Phi = T$).

Finally, $C_1 = C_4$ directly give the value Φ_Γ. This method is not investigated here but has proven to give accurate approximations of gradients at the interface.

8.2.3.3 Comparison of of the normal gradient reconstruction

Finally, the different system presented above are referred as:

- The least square quadratic fit including phase and interface values (LSQ-PI)
- The least square quadratic fit only including phase values (LSQ-P)
- The least square quadratic fit using linear extrapolation (LSQ-LE)
- The least square quadratic fit using quadratic extrapolation (LSQ-QE)
- The least square linear fit using linear extrapolation (LSL-LE)
- The least square linear fit using quadratic extrapolation (LSL-QE)



(a) *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ **(b)** *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$

Figure 8.19: *Mesh convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ *and* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$ *for the different least square approaches using the TFF*

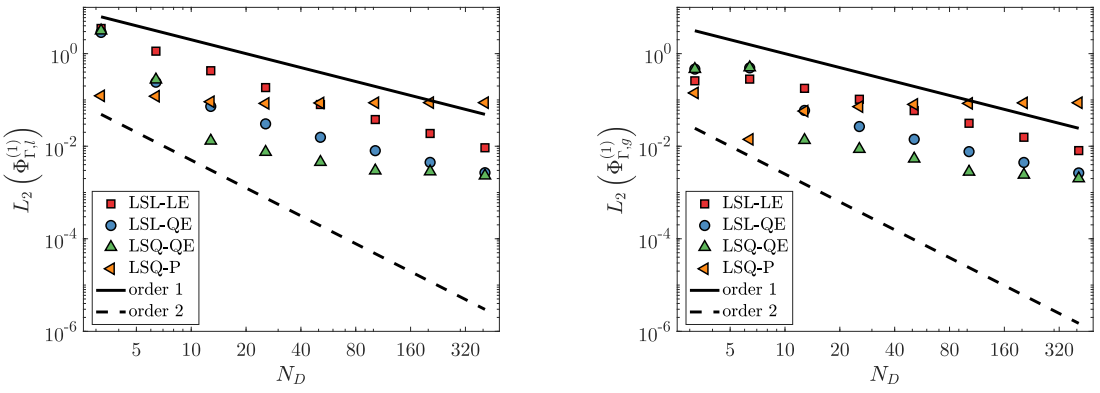
The results are presented in Fig. 8.19 where the LSL fit is first-order accurate while LSQ fit is second-order unless for LSQ-LE and LSQ-PI. As for the results of NI-LE and NI-QE, using LSL fit on the quadratic interpolation improves the accuracy magnitude. LSQ-LE and LSL-LE errors collapse as the slope is already imposed by the linear extrapolation and the second-order derivative is accordingly zero.
LSQ-PI and LSQ-P do not provide the same accuracy for high resolutions. Indeed, LSQ-PI is second-order accurate for coarse to medium meshes while it decreases to first-order convergence for high resolutions while LSQ-P is second-order for all resolutions. This is again explained by the second-order error introduced by the interface location approximation.
Finally, using only one-sided values (LSQ-P) of Φ is more accurate than using the extrapolated values (LSQ-QE) because the one-sided only implies exact values of Φ while LSQ-QE

introduce a third order error on more than half of the values used in the least square minimization.

8.2.4 Application to a second-order scalar field

For completeness, the same gradient methodologies are applied to a second-order scalar field obtained from the static diffusion problem presented in Section 8.3.3. This last study is relevant for evaporation problems where the scalar field is not exact and evolves in time through diffusion and convection. As demonstrated in Section 8.3.3, the diffusion scheme with embedded boundary conditions considered in the solver is second-order accurate at best. Then, a first-order accuracy is expected on gradient evaluation. For a phase-change solver, it means that $\dot{m}$ is evaluated at first-order from the gradient reconstruction.
The most accurate gradient evaluations retained in the above study are directly applied on the scalar field obtained from the static diffusion problem with Dirichlet boundary conditions of Section 8.3.3. This problem is analogous to the resolution of the energy equation with computation of the evaporation rate $\dot{m}$ from $\mathcal{M}_T$.

VOF normal gradient For the TFF, the scalar field $\overline{\Phi}^{diff}$ is between first and second-order. Then, the normal gradient is expected to be at most first-order accurate.



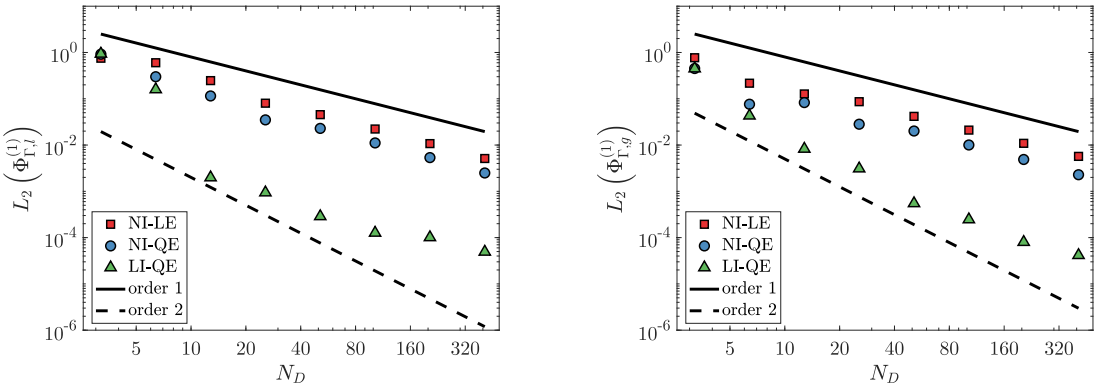
(a) *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ **(b)** *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$

Figure 8.20: *Mesh convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ *and* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$ *for the different least square approaches using TFF applied to* $\overline{\Phi}^{diff}$

The results are presented in Fig. 8.20 for the LSL-LE, LSL-QE, LSQ-QE and LSQ-P methods. The accuracy of the normal computation is drastically decreased compared to the results presented in Section 8.2.3.3. The LSQ-P method is not convergent anymore while other methods show a convergent behaviour. Using the LSL fit provides a solid first-order convergence of the normal with a more accurate computation using quadratic extrapolation. However, the LSQ fit proposed above leads to error saturation for high resolution while it is the most accurate method for low to medium meshes. In fact, trying to fit a quadratic

functional is not the most suited method for its lack of monotonicity as discussed in [122] where an Erf functional was fitted instead. Moreover, a second-order variation captured by such functional from a first to second-order scalar field is not consistent.
From these considerations, a LSL fit will be used either based on linear or quadratic extrapolations and will be referred as VOF-LE and VOF-QE respectively.

SLS normal gradient For the SLS framework, the scalar field $\overline{\Phi}^{diff}$ is second-order accurate with expected first-order normal gradients.



(a) *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ **(b)** *Convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$

Figure 8.21: *Mesh convergence of* $L_2\left(\Phi^{(1)}_{\Gamma,l}\right)$ *and* $L_2\left(\Phi^{(1)}_{\Gamma,g}\right)$ *for the different combination of field extrapolation and gradient interpolation applied to* $\overline{\Phi}^{2nd}$

In Fig. 8.21, the error convergence of NI-LE, NI-QE and LI-QE are represented. All methods have an asymptotic first-order convergence rate while LI-QE is essentially second-order for low to medium resolutions. The use of quadratic extrapolation improves gradient accuracy while performing the additional linear interpolation improves it even more.
In the solver, the NI-QE or LI-QE method are used and will be referred as SLS-LE and SLS-QE respectively.

8.3 Handling of flux jumps at the interface

In a WDF, the jump conditions at the interface are handled with a source terms which require one of the δ_Γ discretizations studied in Section 8.1. When the JCF formulation is used, the jump condition can be ensured either using GFM or by applying a boundary condition at the interface as discussed in Section 7.4.2. Finally, for the TFF, a source term can be imposed at the interface using a δ_Γ discretization or boundary conditions can be applied at the interface.
In the literature, such strategies are usually defined as immersed boundary methods (IBM). In his review, Mittal [137] proposed a distinction between continuous forcing approaches

and discrete forcing approaches.
The continuous forcing approach is due to Peskin [166] for the coupling of elastic boundaries with a flow. In the method, the boundary condition is represented by a set of Lagrangian markers and the coupling is performed through forcing terms located at these makers locations (the analogy can be done with the Front Tracking method used for tracking an interface). As the markers does not belong to the Cartesian grid on which the governing equations are solved, a projection is required to impose the forcing terms to the flow nodes. This procedure implies a smoothing of the forces on several cells through Dirac regularization which removes the sharp nature of the strategy. Moreover, the governing equations need to be solved in the whole domain including the immersed boundaries.
To maintain a sharp framework, the use of discrete forcing approach is necessary. The idea is to modify the stencil near the immersed boundary to impose the associated boundary conditions. In this section, two frameworks are presented to use the interface representation for imposing boundary conditions in interface cells

- The cut-cell method uses explicitly the boundary defined by the interface reconstruction to impose the boundary condition.
- The ghost-cell method defines ghost values in the boundary which are interpolated to implicitly impose the boundary condition at the interface.

While the cut-cell method is based on a finite volume framework more suited for VOF, the ghost-cell method was developed in a finite difference method relying on the signed distance provided by the Level Set for ghost-values interpolation.
This section will only describe the resolution of diffusion problems with boundary conditions at the interface as it is analogous to impose the heat flux for the energy equation or a mass flux for the species mass fraction equation. Given the general Poisson equation defined in Appendix A.1

$$\nabla \cdot (\beta \nabla \Phi) = S \quad . \tag{8.41}$$

This equation is solved separately in Ω_l and Ω_g with boundary conditions at the interface instead of an equation solved in Ω with the jump conditions $[\Phi]_\Gamma = a_\Gamma$, $[\beta \nabla \Phi \cdot \mathbf{n}_\Gamma]_\Gamma = b_\Gamma$. In the following, the general boundary condition can be written as

$$a\Phi_\Gamma + b\Phi_\Gamma^{(1)} = c \quad , \tag{8.42}$$

with a, b and c defined such that

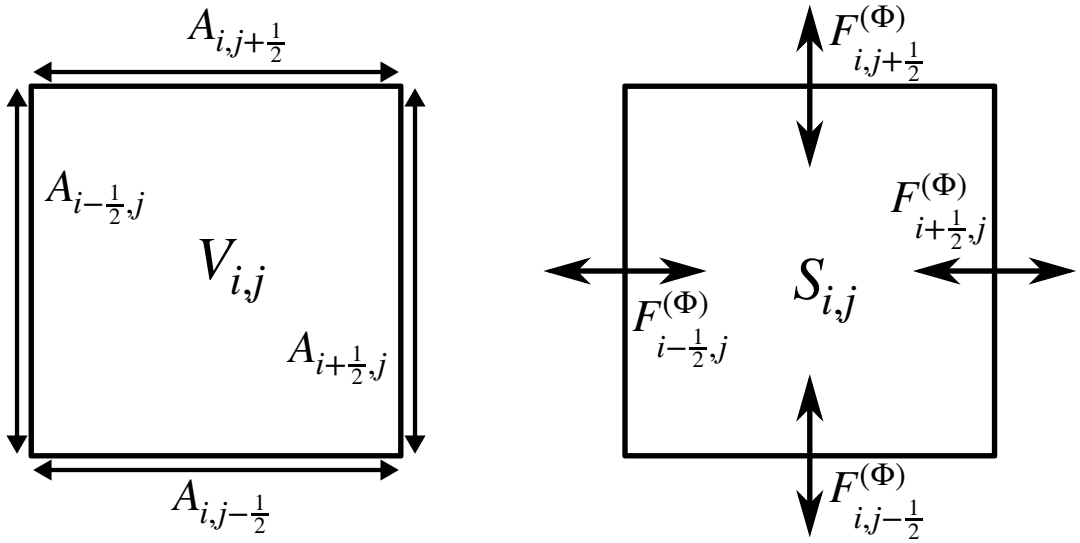
1. If $a \neq 0$ and $b = 0$, it reduces to a Dirichlet boundary condition
2. If $a = 0$ and $b \neq 0$, it reduces to a Neumann boundary condition
3. If $a \neq 0$ and $b \neq 0$, it corresponds to a Robin boundary condition

Note that a, b and c are not known a priori for a phase-change solver application and need to be reconstructed using the methods presented in Section 8.2 for Φ_Γ and $\Phi_\Gamma^{(1)}$ evaluation respectively.

8.3.1 Cut-cell method

The method was first presented as the Cartesian grid method [175] in opposition to body-fitted grids more popular at this time to treat complex geometry. The main advantage of

this strategy is to maintain a uniform discretization in the computational domain while the boundaries are treated with explicit reconstruction of the body crossing the computational cells. It was first applied to the study of compressible flows [165] and then adapted to the resolution of the Poisson equation in 2D [129] and later in 3D [196].
The work in [129] considers the resolution of the heat equation on irregular time-dependant domains.



(a) *Area definition* **(b)** *Fluxes definition*

Figure 8.22: *Illustration of the area and flux definitions for a pure cell*

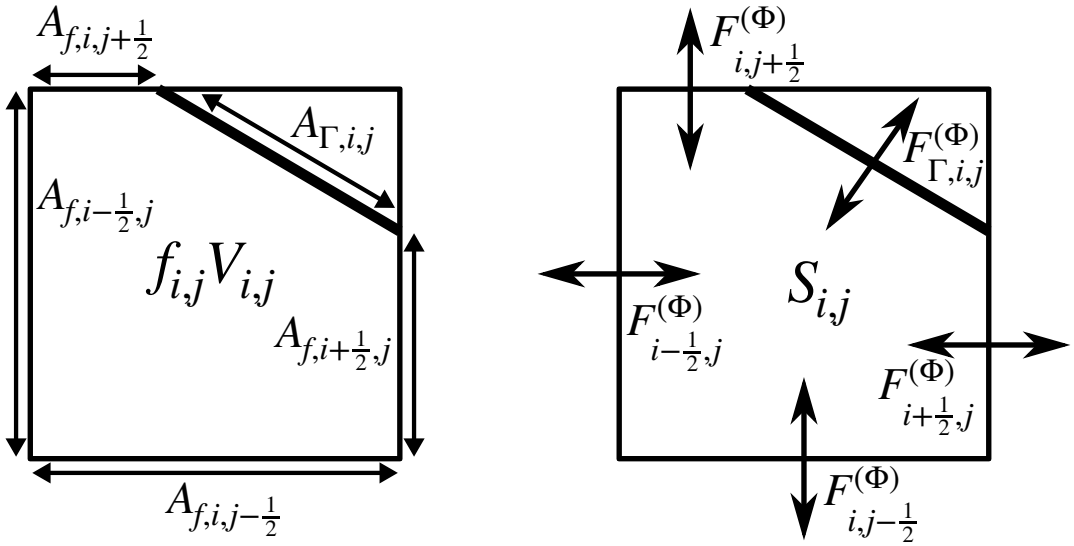
When the cell is not cut by the interface, as illustrated in Fig. 8.22, the following 2D finite volume discretization is proposed for Eq. (8.41)

$$\begin{aligned} &F^{(\Phi)}_{i+\frac{1}{2},j}A_{i+\frac{1}{2},j} - F^{(\Phi)}_{i-\frac{1}{2},j}A_{i-\frac{1}{2},j} \\ &+F^{(\Phi)}_{i,j+\frac{1}{2}}A_{i,j+\frac{1}{2}} - F^{(\Phi)}_{i,j-\frac{1}{2}}A_{i,j-\frac{1}{2}} = S_{i,j}V_{i,j} \quad , \end{aligned} \tag{8.43}$$

with A the face area (length in 2D) and V the cell volume (area in 2D).
In the case of a cell containing the interface, an additional contribution is due to the interface and the configuration of Fig. 8.23 can be discretized by

$$\begin{aligned} &F^{(\Phi)}_{i+\frac{1}{2},j}A_{f,i+\frac{1}{2},j} - F^{(\Phi)}_{i-\frac{1}{2},j}A_{f,i-\frac{1}{2},j} \\ &+F^{(\Phi)}_{i,j+\frac{1}{2}}A_{f,i,j+\frac{1}{2}} - F^{(\Phi)}_{i,j-\frac{1}{2}}A_{f,i,j-\frac{1}{2}} + F^{(\Phi)}_{\Gamma,i,j}A_{\Gamma,i,j} = S_{i,j}f_{i,j}V_{i,j} \quad , \end{aligned} \tag{8.44}$$

with A_f the wetted areas, $F^{(\Phi)}_{\Gamma,i,j}$ the interface flux, $A_{\Gamma,i,j}$ the interface area and $f_{i,j}$ the volume fraction of the known part of the domain in the cell. In Eq. (8.44), the face and interface fluxes need to be evaluated.

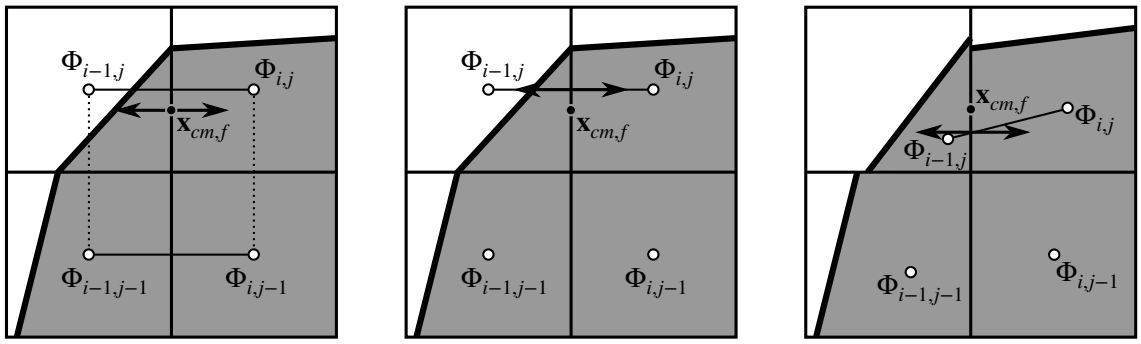


(a) *Area definition* **(b)** *Fluxes definition*

Figure 8.23: *Illustration of the area and flux definitions for a mixed cell*

8.3.1.1 Face flux evaluation

First, the fluxes have to be approximated at the faces $F_f^{(\Phi)}$ where A_f does not necessarily correspond to the computational cell face. An illustration of different approximations is given in Fig. 8.24.



(a) *McCorquodale et al.* **(b)** *Papac et al.* **(c)** *Palmore et al.*

Figure 8.24: *Illustration of the flux approximation $F^{(\Phi)}_{i-\frac{1}{2},j}$ for different cut-cell approaches. Double-arrows represent the flux $F^{(\Phi)}_{i-\frac{1}{2},j}$ while plain lines represent the associate stencil. Dotted lines represent the interpolation between fluxes.*

McCorquodale et al. [129] use an interpolation of face fluxes $F_f^{(\Phi)}$ at the wetted area

barycenter $\mathbf{x}_{cm,f}$ as shown in Fig. 8.24a:

$$F^{(\Phi)}_{i-\frac{1}{2},j} = \beta_{i-\frac{1}{2},j}\left(\frac{1+\theta_f}{2}\frac{\Phi_{i,j}-\Phi_{i-1,j}}{\Delta x} + \frac{1-\theta_f}{2}\frac{\Phi_{i,j-1}-\Phi_{i-1,j-1}}{\Delta x}\right) \quad , \tag{8.45}$$

with $\theta_f = A_{f,i-\frac{1}{2},j}/A_{i-\frac{1}{2},j}$ the edge fraction.
In [160], $F^{(\Phi)}_f$ is expressed at the cell face barycenter (see Fig. 8.24b) using only neighbours of $\mathcal{C}_{i,j}$

$$F^{(\Phi)}_{i-\frac{1}{2},j} = \beta_{i-\frac{1}{2},j}\frac{\Phi_{i,j}-\Phi_{i-1,j}}{\Delta x} \quad . \tag{8.46}$$

This is a first-order approximation of the flux $F^{(\Phi)}_{i-\frac{1}{2},j}$ where the edge fraction θ_f is approximated by $\theta_f = 1$. However, the main motivation for such decrease in accuracy is to maintain a symmetric definition of the outcome matrix which is interesting for efficient resolution of the linear system associated.
Finally, a last definition of $F^{(\Phi)}_f$ proposed by [159] and illustrated in Fig. 8.24c is the following

$$F^{(\Phi)}_{i-\frac{1}{2},j} = \beta_{i-\frac{1}{2},j}\left(\Phi_{i,j}-\Phi_{i-1,j}\right)\frac{\Delta x_{cm}}{\|\mathbf{x}_{cm,i,j}-\mathbf{x}_{cm,i-1,j}\|_2^2} \quad , \tag{8.47}$$

with $\Phi_{i,j}$ defined naturally at the cut-cell barycenter $\mathbf{x}_{cm,i,j}$ and Δx_{cm} the distance in the x direction of both cell barycenters. This last expression is still not at second-order, but allows to keep a symmetric discretization of the gradient. Note that for a cell without interface (as $\mathcal{C}_{i,j-1}$ in Fig. 8.24c), $\mathbf{x}_{cm,i,j-1} = \mathbf{x}_{i,j-1}$.
Also, if the cut-cell is defined from a PLIC reconstruction which is not continuous, $A_{f,i-\frac{1}{2},j}$ need to be defined from $A^-_{f,i-\frac{1}{2}}$ the wetted face area of the PLIC reconstruction from cell $\mathcal{C}_{i-1}$ and $A^+_{f,i-\frac{1}{2}}$ the wetted face area of the PLIC reconstruction from cell $\mathcal{C}_i$. An illustration of this discontinuity is presented in Fig. 8.25.
Several approximations of $A_{f,i-\frac{1}{2}}$ from $A^-_{f,i-\frac{1}{2}}$ and $A^+_{f,i-\frac{1}{2}}$ can be expressed such as the min, max or an average operator $\langle\ \rangle$ to be defined later

$$A_{f,i-\frac{1}{2}} = \min\left(A^+_{f,i-\frac{1}{2}}, A^-_{f,i-\frac{1}{2}}\right) \quad , \tag{8.48}$$

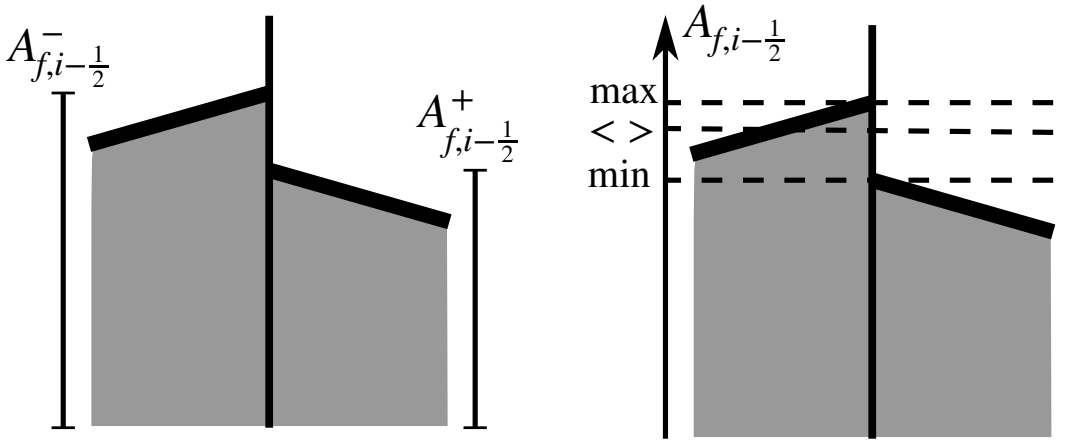
$$A_{f,i-\frac{1}{2}} = \max\left(A^+_{f,i-\frac{1}{2}}, A^-_{f,i-\frac{1}{2}}\right) \quad , \tag{8.49}$$

$$A_{f,i-\frac{1}{2}} = \left\langle A^+_{f,i-\frac{1}{2}}, A^-_{f,i-\frac{1}{2}}\right\rangle \quad . \tag{8.50}$$

In the work of [159], they found that using Eq. (8.48) was providing the most stable results.

8.3.1.2 Interface flux evaluation

Then, the definition of $F^{(\Phi)}_{\Gamma,i,j} = \beta_{\Gamma,i,j}\Phi^{(1)}_\Gamma$ depends on the type of boundary condition that needs to be prescribed at the interface.



(a) *Definition of* $A^-_{f,i-\frac{1}{2}}$ *and* $A^+_{f,i-\frac{1}{2}}$ **(b)** *Different approximation of* $A_{f,i-\frac{1}{2}}$

Figure 8.25: *Illustration of the wetted area approximation from discontinuous PLIC reconstructions at face* $A_{i-\frac{1}{2}}$

Dirichlet boundary condition To apply Dirichlet boundary condition Φ_Γ, one need to compute $F^{(\Phi)}_{\Gamma,i,j}$ from an approximation. This can be done using quadratic interpolation as initially proposed in [129] and used in [162] for contact lines in complex geometries

$$F^{(\Phi)}_{\Gamma,i,j} = \beta_{\Gamma,i,j} \frac{1}{\delta_2 - \delta_1} \left(\frac{\delta_2}{\delta_1} \left(\Phi_\Gamma - \widetilde{\Phi}\left(\delta_1\right) \right) - \frac{\delta_1}{\delta_2} \left(\Phi_\Gamma - \widetilde{\Phi}\left(\delta_2\right) \right) \right) \quad , \tag{8.51}$$

with $\delta_1 = \Delta x$ and $\delta_2 = 2\Delta x$ the normal distance to the interface barycenter $\mathbf{x}_\Gamma$ and $\widetilde{\Phi}$ a quadratic reconstruction of Φ. This has been generalized to 3D in [196].
In [159], a first-order gradient approximation is built from $\mathbf{x}_\Gamma$ and $\mathbf{x}_{cm}$

$$F^{(\Phi)}_{\Gamma,i,j} = \beta_{\Gamma,i,j} \frac{\Phi_i - \Phi_\Gamma}{\|\mathbf{x}_{cm} - \mathbf{x}_\Gamma\|_2} \quad . \tag{8.52}$$

This expression allows to handle the boundary condition fully implicitly while keeping a symmetric matrix.

Neumann boundary condition For Neumann boundary condition, the choice is straightforward and has been used in several works [129; 196; 144; 131]. The flux is directly prescribed by the Neumann condition

$$F^{(\Phi)}_{\Gamma,i,j} = \beta_{\Gamma,i,j} \frac{c_{i,j}}{b_{i,j}} \quad . \tag{8.53}$$

Robin boundary condition The Robin boundary condition is very interesting for evaporation applications and has been presented by [160] for the resolution of the Stefan prob-

lem. This takes the form

$$F^{(\Phi)}_{\Gamma,i,j} = \beta_{\Gamma,i,j}\frac{c_{i,j} - a_{i,j}\Phi_\Gamma}{b_{i,j}} = \beta_{\Gamma,i,j}\frac{c_{i,j} - a_{i,j}\Phi_{i,j}}{b_{i,j}} \quad , \tag{8.54}$$

with the first-order approximation $\Phi_\Gamma = \Phi_{i,j}$.

8.3.2 Ghost cell method

In parallel to the Cut-cell method, another approach has been developed in the late 90's to handle Poisson equations on irregular domains.

8.3.2.1 Dimensional-splitting Dirichlet boundary condition

The first algorithm was proposed in [34] to apply Dirichlet boundary conditions at the interface. Then, other works such as [68] proposed an extension of GFM to impose boundary conditions instead of quantity jumps at the interface.
The following second-order finite difference discretization is proposed for the 1D version of Eq. (8.41) where the interface lies between x_i and x_{i+1}

$$\frac{\beta_{i+\frac{1}{2}}\frac{\Phi^{gh}_{i+1}-\Phi_i}{\Delta x} - \beta_{i-\frac{1}{2}}\frac{\Phi_i-\Phi_{i-1}}{\Delta x}}{\Delta x} = S_i \quad , \tag{8.55}$$

with Φ^{gh}_{i+1} the ghost value of Φ in the unknown domain.
Then, Φ^{gh}_{i+1} can be defined following different level of accuracy.

- A first-order approximation $\Phi^{gh}_{i+1} = \Phi_\Gamma$
- A second-order approximation $\Phi^{gh}_{i+1} = \frac{\Phi_\Gamma+(\theta-1)\Phi_i}{\theta}$
- A third order approximation $\Phi^{gh}_{i+1} = \frac{2\Phi_\Gamma+(2\theta^2-2)\Phi_i+(-2\theta^2+1)\Phi_{i-1}}{\theta^2+\theta}$
- A fourth order approximation (details provided in [67])

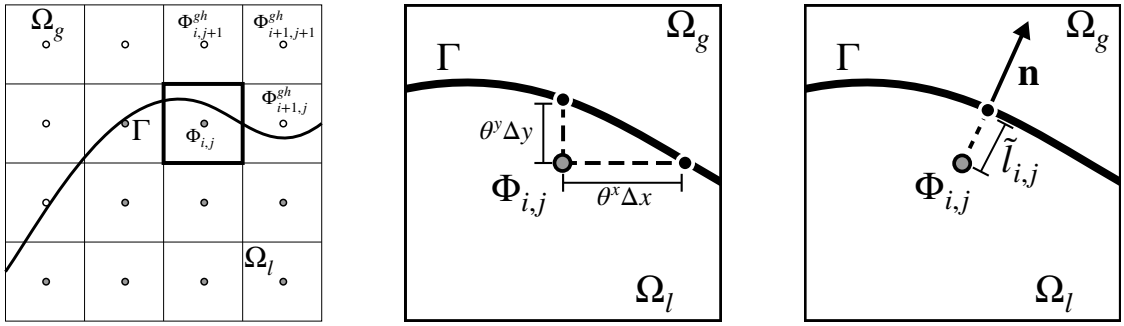
corresponding respectively to the constant, linear, quadratic and cubic extrapolation of Φ at x_Γ only using values of the known domain with $\theta = |\phi_i|/(|\phi_{i+1}| + |\phi_{i+1}|)$. The quadratic extrapolation was already used for the resolution of Stefan problem in the Level Set context [34] but leads to a non-symmetric linear system. This is also the case for the cubic interpolation used in [67] to achieve fourth order accuracy of the Laplace and Heat equations.
As for the cell volume in the cut-cell method, the expressions including θ are not well-defined for arbitrary small values of θ. This issue can be handled by fixing a threshold θ_c for which the interface x_Γ is too close to x_i and imposing $\Phi_i = \Phi_\Gamma$ does not alter the overall accuracy and boundness of the method. Usually, the threshold is chosen as $\theta_c = \Delta x$. Another way to handle such singularity is to use a new stencil which does not include Φ_i in the discretization [7]. Then the denominator is bounded by Δx.
The most popular choice is the linear extrapolation which leads to a second-order accurate scheme

$$\frac{\beta_{i+\frac{1}{2}}\frac{\Phi_\Gamma-\Phi_i}{\theta\Delta x} - \beta_{i-\frac{1}{2}}\frac{\Phi_i-\Phi_{i-1}}{\Delta x}}{\Delta x} = S_i \quad . \tag{8.56}$$

The convergent behaviour was first observed numerically without any evidence from Taylor expansion analysis, however in [94] a theoretical proof is provided.
This approach is easily extended to multidimensional Poisson equation by applying the same methodology for the other directions as depicted in Fig. 8.26b. However, application to other boundary conditions such as Neumann or Robin is not straightforward in the finite difference framework and has been addressed only recently.

8.3.2.2 Normal Robin boundary condition

The main obstacle to develop Neumann and Robin boundary conditions is their normal direction nature. In [30], authors demonstrated that using a dimensional-splitting is inherently first-order and not even consistent for the gradient evaluation.



(a) *Schematic of GCM* **(b)** *Dimensional-splitting* **(c)** *Normal direction*

Figure 8.26: *Illustration of the GCM applied to cell $\mathcal{C}_{i,j}$ in both dimensional-splitting and normal direction implementations*

They propose a new procedure for imposing a Robin boundary condition in the normal direction to the interface using a ghost-cell method.
First, the extrapolation procedure described above for Dirichlet boundary condition is generalized for Robin boundary condition. The linear extrapolation functional $\widetilde{\Phi}$ can be rewritten as

$$\widetilde{\Phi}(\widetilde{x}) = m\widetilde{x} + n \quad , \tag{8.57}$$

with $\widetilde{x}$ the position coordinate in the normal direction to the interface such that the interface is located at $\widetilde{x}_\Gamma = 0$. The unknown m and n are retrieved by using the following system

$$\begin{cases} \widetilde{\Phi}(-\theta\Delta x) = -m\theta\Delta x + n \\ \widetilde{\Phi}(0) = n \\ \widetilde{\Phi}^{(1)}(0) = m \end{cases} . \tag{8.58}$$

This system is overdetermined and needs to be reduced to a system of only two equations. This is done by plugging m and n in Eq. (8.42)

$$m = \frac{c - an}{b} \tag{8.59}$$

Instead of directly applying the values $\Phi(0) = n$ and $\Phi^{(1)}(0) = m$, the simple relation Eq. (8.59) is used to obtain the following system

$$\begin{cases} \widetilde{\Phi}(-\theta\Delta x) = -m\theta\Delta x + n \\ m = (c - an)/b \end{cases} . \tag{8.60}$$

Then the value at Φ_{i+1}^{gh} does not depend explicitly on the quantities at the interface. Solving Eq. (8.60) leads to

$$m = \frac{c - a\Phi_i}{b + a\theta\Delta x} \qquad \text{and} \qquad n = \frac{b\Phi_i + c\theta\Delta x}{b + a\theta\Delta x} \quad , \tag{8.61}$$

the value Φ_{i+1}^{gh} is then written as

$$\Phi_{i+1}^{gh} = \widetilde{\Phi}((1-\theta)\Delta x) = \Phi_i + \frac{(c - a\Phi_i)\Delta x}{b + a\theta\Delta x} \quad . \tag{8.62}$$

The generalization to arbitrary order of accuracy is obtained by adding more points in the stencil. For details on the implementation, see [30].
This above system can be generalized in multi-dimensional cases by splitting the boundary condition Eq. (8.42) into two (or three) boundary equations projected on the Cartesian directions. Then, each direction is treated separately. As pointed out before, such treatment is inherently first-order.
To treat the problem directly in the normal direction, the linear system of Eq. (8.60) is solved in the normal direction by introducing a normal distance $\widetilde{l}_{i,j}$ between the interface $\mathbf{x}_\Gamma$ and the known point $\mathbf{x}_{i,j}$ as illustrated in Fig. 8.26c. In most of the case $\widetilde{l}_{i,j} \neq \theta^x \Delta x$ and $\widetilde{l}_{i,j} \neq \theta^y \Delta y$.
The following algorithm is used to define the ghost values in a narrow band around the boundary

1. In cells close to the interface, solve Eq. (8.60) to retrieve the normal derivative $m = \Phi^{(1)}$ while compute $\Phi^{(1)}$ from standard second-order differences for the other cells in the known domain.
2. Extrapolate the $\Phi^{(1)}$ field using an extrapolation procedure of Appendix B.1
3. Extrapolate the Φ field by solving Eq. (B.3)

Then the standard discretization in Eq. (8.55) is used for the Laplacian with Φ_{i+1}^{gh} defined by the procedure above.

8.3.2.3 General boundary condition

Finally, the procedure to impose Robin boundary conditions has been extended to the general boundary condition of Eq. (8.42) in [31]. It allows to impose boundary conditions in the normal direction to the interface.
The algorithm step 1 can be adapted to Dirichlet boundary condition by fixing $b = 0$ in Eq. (8.61), then the normal derivative is defined as

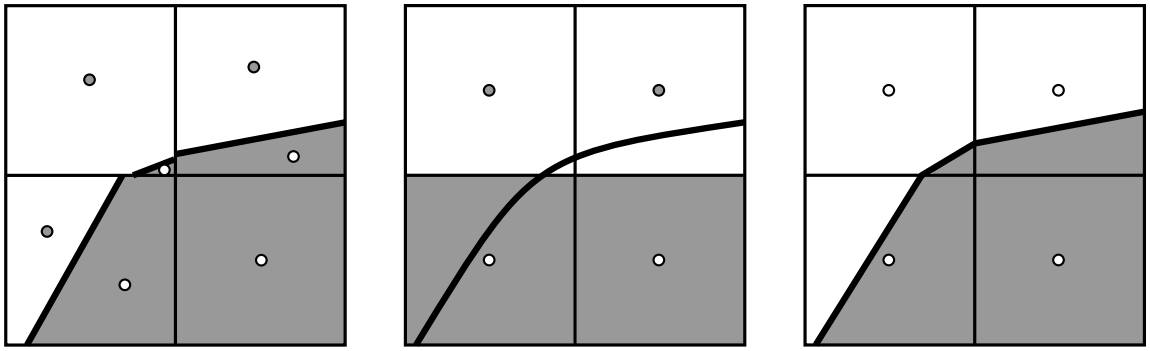
$$\Phi^{(1)} = \frac{c - a\Phi_i}{a\widetilde{l}_{i,j}} \quad . \tag{8.63}$$

In this expression, the derivative is of arbitrary high magnitude when $\widetilde{l}_{i,j}$ is close to zero. In such case, $\Phi^{(1)}$ is computed from second-order central differences.
For a Neumann boundary condition, $a = 0$ in Eq. (8.61) leading to

$$\Phi^{(1)} = \frac{c}{b} \quad . \tag{8.64}$$

8.3.3 Application to static diffusion problems with flux jump

In this section, a 2D static diffusion problem is considered to compare WDF, JCP and TFF relying on VOF or SLS representation of the interface. For the WDF, a source term is applied, while for the JCP and TFF, it is alos possible to impose boundary conditions at the interface using either CCM or GCM (represented here in Fig. 8.27).



(a) *VOF-CCM* **(b)** *JCF* **(c)** *SLS-CCM*

Figure 8.27: *Scalar definition for different representations of the interface. The white and grey areas are the gas and liquid phase respectively. The white and grey points are the liquid and gas scalar position respectively.*

The study is based on the test case used for the quantities reconstruction in Section 8.2. The associated static diffusion problem is written as

$$\nabla \cdot \nabla \Phi = \begin{cases} \left(4r^2 - 4\right) \exp\left(-R^2\right) & \text{if } r > R \\ \Phi = -40 & \text{if } r \leq R \end{cases} \quad , \tag{8.65}$$

with either a flux jump, a Dirichlet, Neumann or Robin boundary condition defined respectively as

$$[\nabla \Phi \cdot \mathbf{n}_\Gamma]_\Gamma = -20R + 2R \exp\left(-R^2\right) \quad , \tag{8.66}$$

$$\Phi_\Gamma = 0 \quad , \tag{8.67}$$

$$\Phi^{(1)}_{\Gamma,l} = -20R \quad , \tag{8.68}$$

$$\Phi^{(1)}_{\Gamma,g} = -2R \exp\left(-R^2\right) \quad . \tag{8.69}$$

Note that the Robin boundary is considered imposing $a = c$ and $b = 1$ for analogy with the Robin boundary condition associated to Y given by Eq. (7.70).

The error is then computed using

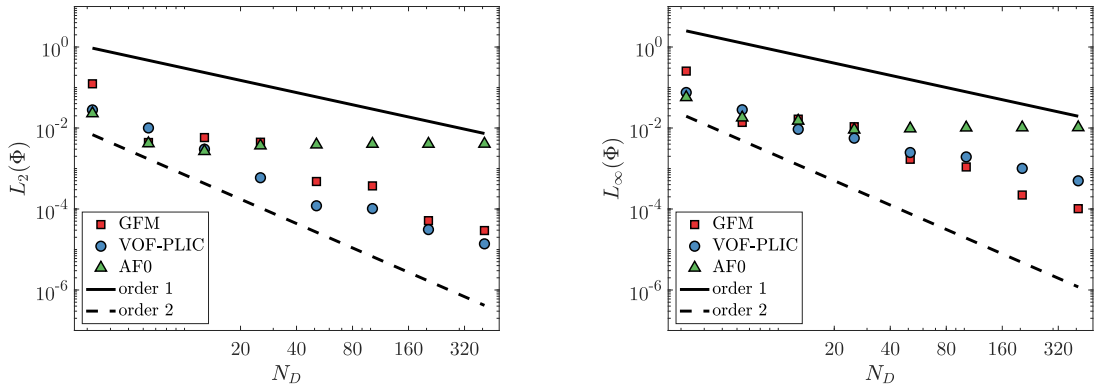
$$L_2(\Phi) = \sqrt{\frac{1}{N_\Omega}\sum_{i=1}^{N_\Omega}\left(\Phi_{i,exact} - \Phi_i\right)^2} \quad , \tag{8.70}$$

$$L_\infty(\Phi) = \max_i \left|\Phi_{i,exact} - \Phi_i\right| \quad , \tag{8.71}$$

with N_Ω the number of points in the domain. Note that for the TFF, there are two points in the mixed cells, one belonging to Ω_l and one to Ω_g.

8.3.3.1 WDF approach

This section considers the WDF where the static diffusion problem Eq. (8.65) is solved in the whole domain by imposing Eq. (8.66). The problem is analogous to the resolution of T or P with a flux jump at the interface $[k\nabla T \cdot \mathbf{n}_\Gamma]_\Gamma = \dot{m}\,[h]_\Gamma$ or $\left[\frac{1}{\rho}\nabla P \cdot \mathbf{n}_\Gamma\right]_\Gamma = [\mathbf{u}\cdot\mathbf{n}_\Gamma]_\Gamma$ respectively. As explained in Section 7.4.1.1, this jump condition can be treated using different Dirac definitions. Here, the GFM, the algebraic formulae AF0 defined in Section 8.1.1 and the VOF-PLIC surface approximations are used to define the Dirac distribution through Eq. (7.10).



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

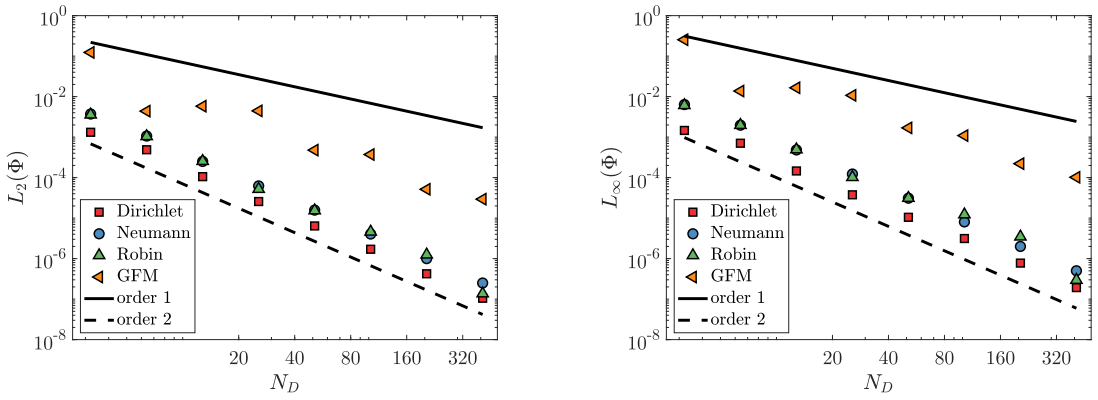
Figure 8.28: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the static diffusion problem using a WDF approach*

In Fig. 8.28, the convergence of $L_2(\Phi)$ and $L_\infty(\Phi)$ for a WDF. The results show a first to second-order convergence of the error for GFM and VOF-PLIC which are very sharp while an approach using a smoothed source term instead does not lead to a convergent methodology.
The next section aims to show the difference of accuracy obtained for a JCF.

8.3.3.2 JCF approach

In the JCF approach based on GCM for embedded boundaries, the scalar is defined at the cell center. Then, a node is considered to be in the liquid if $\phi > 0$ and in the gas if $\phi \leq 0$ as illustrated in Fig. 8.27b. The original discretization of a Dirichlet boundary condition is based on the work of [68] using a dimensional-splitting. For the Neumann or Robin boundary condition, the boundary condition is inherently normal as discussed in Section 8.3.2 and the method presented in [160] is used instead relying on a finite-volume discretization based on the explicit reconstruction of the interface. The methodology presented in [4] is defined here as the original discretization for Neumann and Robin boundary conditions with the interface reconstruction based on the CD2 method. However, this method needs to reconstruct the interface explicitly, and the definition domain is not clear anymore as some interface cells could lye in a different domain which is not compatible with the JCF representation of the interface. In Fig. 8.27c, the two top cells need to be included in the liquid linear system even if they actually lie in the gas phase. The domain of definition then depends on the boundary condition that needs to be applied which is not necessarily desirable.
To keep a JCF scalar definition for any boundary condition, the normal finite difference discretization presented in [30] is investigated and compared to the original discretization. Finally, all this boundary condition methodologies are compared to GFM to quantify the gain or loss in accuracy associated to this method choice.

Original discretization As presented in the literature, all those methods achieve second-order convergence for all type of boundary conditions as presented in Fig. 8.29. The discretization using the boundary conditions at the interface is more accurate than the original GFM method. Hence, the use of boundary conditions have to be prioritized when the quantities of Φ_Γ or $\Phi^{(1)}$ are available (this is not always the case).

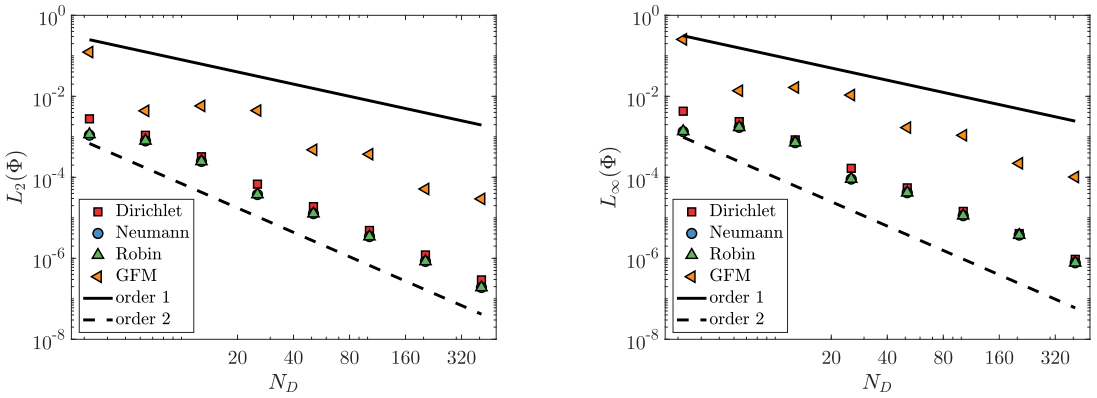


(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

Figure 8.29: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the static diffusion problem using the original JCF approach*

Note that Neumann and Robin boundary conditions require the explicit reconstruction of the interface using the CD2 method. This implies an important increase in computational cost.

Ghost-cell method Another discretization is proposed where all boundary condition types are treated using the general methodology presented in Section 8.3.2. In Fig. 8.30, the mesh convergence of this alternative methodology is presented. Second-order convergence is also met with comparable error magnitude as for the original discretization.



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

Figure 8.30: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the static diffusion problem using the normal finite difference JCF*

More specifically, the Dirichlet boundary condition is less accurate than the dimensional-splitting approach while Neumann and Robin boundary conditions are as accurate as the one obtained using the cut-cell method.
Then, it is more interesting to use the ghost-cell method for Neumann or Robin boundary conditions as it does not require to reconstruct the interface while keeping a unique representation of the scalars regardless the embedded boundary condition.

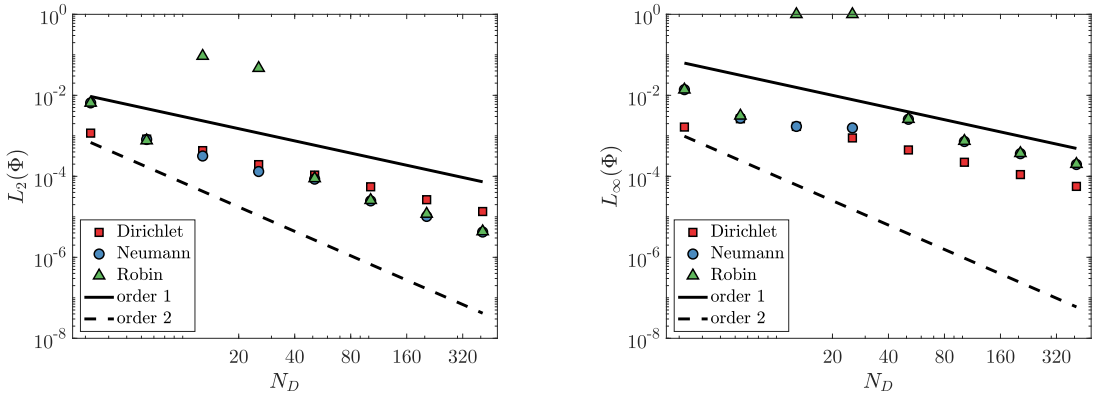
8.3.3.3 TFF approach

The TFF discretization considered here uses the PLIC reconstruction to localize the interface position and defines the phase scalar at its phase barycenter as illustrated in Fig. 8.27a. This means that mixed cells contain both the liquid and gas information without any numerical mixing. The challenge then relies in a proper definition of the fluxes in a truncated control volume when:

- The interface representation is not continuous
- The phase barycenter location is different from the cell center

A first discretization is investigated based on the work of [159]. Then, some adjustments are proposed to enhanced the stability and accuracy of the method.

Original discretization The method proposed in [159] uses the flux discretization of Eq. (8.47) and the wetted areas are obtained with Eq. (8.48). Using such discretization with either Dirichlet, Neumann or Robin lead to the following $L_2(\Phi)$ and $L_\infty(\Phi)$ errors convergence displayed in Fig. 8.31.



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

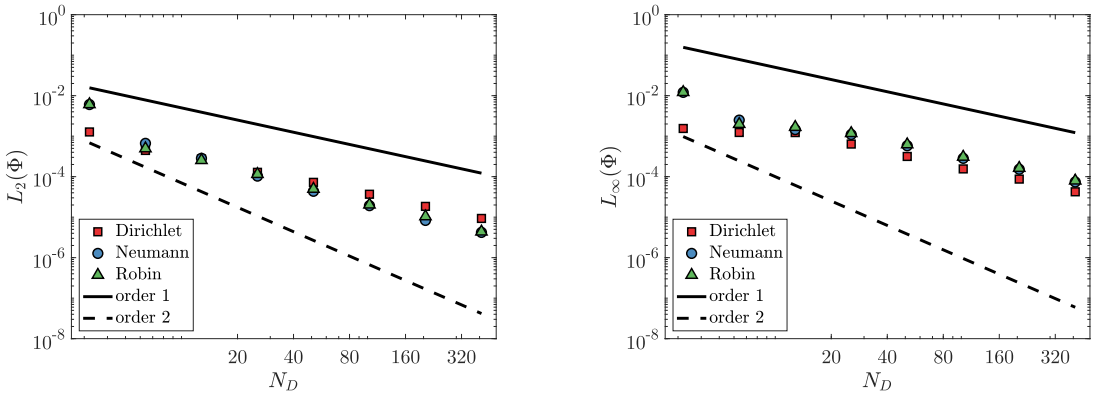
Figure 8.31: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the static diffusion problem using using the original TFF method*

The Dirichlet methodology leads to a first-order convergence which is due to the first-order flux approximation at the interface from Eq. (8.52). When a Neumann boundary condition is applied, the flux at the interface is exact and the method is between first and second-order. second-order accuracy is still not achieved because of the face flux approximation based on non-uniform evaluation points. The Robin boundary condition is slightly less accurate than the Neumann boundary condition which is expected as at is a mixing between Dirichlet and Neumann. However, some points leads to inconsistent results. After some investigations, this is due to some mixed cells with all wetted face areas set to zero because of the approximation Eq. (8.48), then the linear system lead to the simple expression $a\Phi = c$. As we set $a = c$, this reduces to $\Phi = 1$ which is inconsistent in most of the cases.
To improve the accuracy and avoid any inconsistent linear systems, a modified discretization is proposed in this work.

Improvement of the discretization Here, the wetted areas are defined from a weight average of both PLIC reconstruction wetted areas

$$A_{f,i-\frac{1}{2}} = \frac{W_{i-1,j}A^-_{f,i-\frac{1}{2}} + W_{i,j}A^+_{f,i-\frac{1}{2}}}{W_{i-1,j} + W_{i,j}} \quad , \tag{8.72}$$

with $W_{i,j} = H^\epsilon_{\Gamma,i,j}(1 - H^\epsilon_{\Gamma,i,j})$. This interpolation is analogous to the curvature interpolation of Eq. (5.74) and gives more importance on the wetted areas computed from interface cells close to 0.5. These cells contain large interface areas which are expected to provide interface reconstructions less prompt to errors.



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

Figure 8.32: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the static diffusion problem using the proposed improved TFF method*

In Fig. 8.32 the $L_2(\Phi)$ and $L_\infty(\Phi)$ errors convergence do not show the same inconsistency for the Robin boundary condition. This is due to the new evaluation of wetted areas proposed in Eq. (8.72) which avoids the symptomatic case met in the previous study.
Even with these improvements, the method for applying Dirichlet boundary conditions is still first-order while the Neumann and Robin boundary conditions provide an accuracy between first and second-order. To reach second-order, the fluxes close to the interface would require additional interpolations. However, this would lead to non-symmetric discretizations with extended stencils which are not ideal for an implicit treatment.
For all these considerations, the improved discretization is used for the VOF framework.

8.3.3.4 Method choice recap for IBM prescription

For SLS methods, the GFM leads to less accurate evaluation of Φ compared to applying embedded boundaries. Moreover, there are no relevant differences between CCM and GCM applied to Neumann or Robin boundary conditions. Then, the choice to use the ghost-cell method for SLS avoids reconstructing explicitly an interface, which is significantly more expensive. It also allows to keep the JCF for all scalars when applying different boundary conditions.
For a Dirichlet boundary condition, applying the original dimensional-splitting discretization of [68] leads to more accurate results and is used in this work.
For VOF methods, the WDF leads to less accurate evaluation of Φ even if the convergence rate is higher. It is also important to notice that a WDF does not provide information of both phases in the mixed cells while the TFF considered here for VOF does. Then, the cut-cell method is used for VOF in the solver with a TFF.

8.4 Conclusion

In this chapter, the reconstructions of interface quantities used in phase-change simulations are meticulously investigated through a wide range of numerical experiments. This numerical study leads to important conclusions on the choice of numerical methods to treat the energy and species equations along with the computation of the evaporation rate:

- VOF and SLS can provide equivalent accuracy by using different numerical strategies. However, SLS seems to provide the most accurate quantity reconstruction at the interface.
- The evaporation rate needs to be defined carefully at the position of the interface. The least-square fit employed in VOF needs to be centered on the interface while an additional interpolation procedure is required for SLS.
- The resolution of diffusion with embedded boundaries is limited to a one to second-order accuracy for VOF and to second-order for SLS. This is the limiting accuracy for the evaporation rate reconstruction which can achieve first-order accuracy at best.

The resulting choices are summed up in Table 8.2

method	VOF	SLS
δ_Γ	PLIC	GFM
$\dot{m}$	LSL	GCM
IBM	CCM	GCM

Table 8.2: *Summary of computation choices for VOF and SLS*

In the next chapter, the unified framework presented in Chapter 5 is extended to phase change accordingly to the above remarks. The phase-change solver feasibility is then demonstrated on various evaporation test cases from canonical planar Stefan problem to 3D convected droplet.
In the manuscript, the study is restricted to VOF and SLS interface capturing methods even if most of the points raised in this chapter can be transposed to adapt the unified framework to ACLS and CLSVOF.

Chapter 9

Simulation of droplet evaporation

Contents

This final chapter aims to demonstrate the feasibility of accurate evaporation simulations by using all the conclusions from the literature and the numerical experiments of Chapter 8. First, the extension of the unified framework to phase change is presented in Section 9.1 with details on the discretization of the different integrations. Then, numerical results are presented in Section 9.2. They first rely on planar Stefan flow problems, which remove the numerical errors due to topological sources. This set of numerical experiments is presented to evaluate the computation accuracy on T_Γ, Y_Γ and $\dot{m}$, the coupling with the energy and species mass fraction equations. Then, a spherical Stefan flow problem is presented on the 3D cartesian grid to quantify the effects of topology inaccuracies on the phase-change results. Finally, Section 9.3 presents the convection of an evaporating droplet. This last test case allows validating the solver on a fully-coupled configuration where convection, surface tension, diffusion and phase change occurs simultaneously, which is quite challenging.

9.1 Extension of the unified framework to phase change

The unified framework presented in Section 6.1 is now augmented to take into account phase change. The method choices are discussed based on the state-of-the-art presented in Chapter 7 and the preliminary numerical experiments presented in Chapter 8. As for the unified framework without phase change, efforts have been made to keep similar discretizations for VOF and SLS as much as possible. However, some quantities are computed differently depending on the interface capturing method, when using the same approximations is a too important sacrifice in accuracy.

9.1.1 Pressure equation

The pressure equation needs to include the source term due to the continuity equation, this is done using a WDF for VOF with the following discretization of Eq. (7.7)

$$\nabla_c \cdot \left(\frac{1}{\rho_f^{n+1}} \nabla_f P^{n+1} \right) = \frac{1}{\Delta t} \left(\nabla_c \cdot \mathbf{u}^* - \dot{s}_\rho^{n+1} \right) \quad , \tag{9.1}$$

with $\dot{s}_\rho^{n+1}$ defined as

$$\dot{s}_\rho^{n+1} = -\dot{m}^{n+1} \left[\frac{1}{\rho} \right]_\Gamma \frac{A_\Gamma^{n+1}}{V_\mathcal{C}} \tag{9.2}$$

with both $\dot{m}^{n+1}$ and A_Γ^{n+1} evaluated using the advanced energy and interface. For SLS, the information A_Γ^{n+1} is not available unless an explicit reconstruction is performed. Here, a GFM approach is prefered to discretize the velocity jump which leads to a comparable area approximation as demonstrated in Section 8.1.4. Then, $\dot{s}_\rho^{n+1}$ is obtained with

$$\dot{s}_\rho^{n+1} = - \left[\frac{1}{\rho} \right]_\Gamma \mathbf{n}_\Gamma \cdot \nabla_c \left(H_{\Gamma,f}^{0,n+1} \dot{m}_f^{n+1} \right) \quad . \tag{9.3}$$

$H_{\Gamma,f}^{0,n+1}$ computation is based on the linear face mapping ϕ_f presented in Section 8.1.1 and $\dot{m}_f^{n+1}$ is also mapped to the face using linear interpolation. Note that the mapping type of $\dot{m}_f^{n+1}$ does not have a huge impact on the results as it is already defined at the interface when computed from the methodology presented in Section 9.1.4.1. Then a more sophisticated mapping such as Eq. (7.22) is unnecessary.

9.1.2 Interface transport

Following the discussion of Section 7.3, the interface is transported using the interface velocity directly to keep a straightforward implementation of the transport algorithm. The algorithms defined in Section 4.1.1 and Section 4.1.2 are used for VOF and SLS respectively. However, a slight modification of the SLS algorithm is given to take into account the

divergence of $\mathbf{u}_\Gamma$. The following equation is solved instead of Eq. (4.2)

$$\phi_{i,j,k}^{n+1} = \phi_{i,j,k}^{n} - \frac{\Delta t}{\Delta x}\left(F^{(\phi)}_{i+\frac{1}{2},j,k} - F^{(\phi)}_{i-\frac{1}{2},j,k}\right) \tag{9.4}$$

$$- \frac{\Delta t}{\Delta y}\left(F^{(\phi)}_{i,j+\frac{1}{2},k} - F^{(\phi)}_{i,j-\frac{1}{2},k}\right) \tag{9.5}$$

$$- \frac{\Delta t}{\Delta z}\left(F^{(\phi)}_{i,j,k+\frac{1}{2}} - F^{(\phi)}_{i,j,k-\frac{1}{2}}\right) \tag{9.6}$$

$$+ \phi_{i,j,k}^{n} \, \nabla_c \cdot \mathbf{u}_\Gamma^n|_{i,j,k} \quad . \tag{9.7}$$

The interface velocity is computed from the liquid velocity as

$$\mathbf{u}_\Gamma^n = \mathbf{u}_l^n - \frac{\dot{m}^n}{\rho_l}\mathbf{n}_\Gamma \quad . \tag{9.8}$$

According to Section 7.3.1, the computation of $\mathbf{u}_l^n$ needs special attention to be divergence-free. This aspects is crucial in our unified framework based on finite-volume methods where divergence errors are even more visible. Here, the Stefan flow cancellation presented in Section 7.3.1.4 is used by solving the following Poisson equation

$$\nabla_c \cdot \left(\frac{1}{\rho_f^n}\nabla_f \Psi\right) = -\dot{s}_\rho^n \quad , \tag{9.9}$$

with $\dot{s}_\rho^n$ discretized either using Eq. (9.2) or Eq. (9.3) depending on the interface capturing method. Then, the liquid velocity used in the construction of $\mathbf{u}_\Gamma$ is retrieved with

$$\mathbf{u}_l^n = \mathbf{u}^n - \frac{1}{\rho_f^n}\nabla_f \Psi \tag{9.10}$$

9.1.3 Energy and species

As demonstrated in Section 8.3.3, the use of a WDF for scalars leads to less accurate results for the diffusion operator. Then, the TFF is used to solve the energy and species equations based on the CCM of Section 8.3.3.3, while the JCF based on the GCM of Section 8.3.3.2 for VOF and SLS respectively. This allows to simplify the equations by considering constant ρ, k, c_p and $\mathcal{D}_v$ in a given phase. The energy and species equations are accordingly simplified into three scalars T_l, T_g and Y transported in their respective phase and coupled through the boundary condition at the interface. The general transport equation for a scalar Φ belonging to a phase $p = l, g$ is given by

$$\frac{\Phi_p^{n+1} - \Phi_p^n}{\Delta t} = \mathcal{L}_{\text{conv}}^n\left(\Phi_p\right) + \mathcal{L}_{\text{diff}}^{n+1}\left(\Phi_p\right) \quad . \tag{9.11}$$

In this discretization, $\mathcal{L}_{\text{conv}}^n\left(\Phi_p\right)$ is treated explicitly while $\mathcal{L}_{\text{diff}}^{n+1}\left(\Phi_p\right)$ is treated implicitly to alleviate numerical issues due to the stiff terms arising from interface conditions at the interface. Moreover, no additional operator is required for the phase change as it is directly taken into account when solving Eq. (9.11) with prescribed boundary conditions at the interface.

9.1.3.1 Convection term $\mathcal{L}_{\mathbf{conv}}(\Phi_p)$

The convection operator is treated using a methodology analogous to Section 5.2.3 proposed in [159]. The idea is to keep consistency between mass and scalar transport. This choice is made for simplicity of implementation and adaptability to any interface capturing method. This leads to the following convection operator:

$$\mathcal{L}^n_{\text{conv}}(\Phi_p) = \frac{1}{\alpha^*_p}\nabla_c \cdot \left(\alpha^n_p \Phi^n_p \mathbf{u}^n_p\right) - \frac{\Phi^n_p}{\Delta t}\frac{\alpha^*_p - \alpha^n_p}{\alpha^*_p} \quad . \tag{9.12}$$

The additional term in the RHS is a correction of scalar due to the mass change in the cell during the timestep. The predicted staggered phase volume fraction α^*_p is obtained from the following continuity equation

$$\frac{\alpha^*_p - \alpha^n_p}{\Delta t} = -\nabla_c \cdot \left(\alpha^n_p \mathbf{u}^n_p\right) \quad . \tag{9.13}$$

Note that α^*_p is only an intermediate evolution variable discarded after applying the convection operator. The initial phase volume fraction α^n_p needs to be deduced from $\alpha^n_p = H^\epsilon_{\Gamma,p}(c^n)$.

$$H^\epsilon_\Gamma(f) = f \quad , \tag{9.14}$$

$$H^\epsilon_\Gamma(\phi) = \frac{1}{2}\left(\tanh\left(\frac{\phi}{2\epsilon}\right)\right) \quad . \tag{9.15}$$

This leads to a smoother version of the effective volume fraction f in the case of SLS, while it is exactly the volume fraction of fluid in the case of VOF. For the same reasons as for the momentum conservation presented in Section 6.2.2, it is expected to have better conservation properties for VOF than for SLS.
The fluxes $F^{(\alpha_p)}$ and $F^{(\alpha_p\Phi_p)}$ have to be evaluated with the same interpolation for consistency. Moreover, the schemes have to be bounded to avoid any overshoot or undershoot of phase volume fraction. In the solver, a BHOUC5 interpolation is performed, which switches to an upwind evaluation when the stencil crosses the interface.

9.1.3.2 Diffusion term $\mathcal{L}_{\mathbf{diff}}(\Phi_p)$

In the VOF framework, the use of VOF-PLIC reconstruction with a CCM is between first and second-order accuracy while the GCM exhibits a second-order accuracy for SLS as presented in Section 8.3.3.
The discretization is reminded for both VOF and SLS frameworks:

- The diffusion operator for VOF based on CCM can be written as

$$\mathcal{L}^{n+1}_{\text{diff}}(\Phi_p) = \frac{1}{\alpha^{n+1}_k}\left(\nabla^x_c\left(\beta_p \frac{A^{n+1}_{f,p}}{A_f}\nabla^x_{f,p}\Phi^{n+1}_p\right) + \nabla^y_c\left(\beta_p \frac{A^{n+1}_{f,p}}{A_f}\nabla^y_{f,p}\Phi^{n+1}_p\right) + \frac{A_\Gamma}{V_\mathcal{C}}F^{(\Phi_p)}_\Gamma\right) \quad , \tag{9.16}$$

with $A_{f,p}^{n+1}$ the wetted area defined with the modified average of Eq. (8.72), A_f the cell face area, A_Γ the interface area and $V_\mathcal{C}$ the cell volume. β_p is the diffusion related to the phase scalar Φ_p. $F_\Gamma^{(\Phi_p)}$ is the interface flux associated to the embedded boundary condition prescribed at the interface. $\nabla_{f,p}^x$ and $\nabla_{f,p}^y$ the non-uniform gradient operators defined at the wetted face barycenter as (here for $\nabla_{f,p}^x \Phi\Big|_{i-\frac{1}{2},j}$)

$$\nabla_{f,p}^x \Phi_p\Big|_{i-\frac{1}{2},j} = \left(\Phi_{p,i,j} - \Phi_{p,i-1,j}\right) \frac{\Delta x_p}{\|\mathbf{x}_{p,i,j} - \mathbf{x}_{p,i-1,j}\|_2^2} \quad , \tag{9.17}$$

with $\mathbf{x}_p$ the phase barycenter and Δx_p the displacement in the x-direction between $\mathcal{C}_{i-1,j}$ and $\mathcal{C}_{i,j}$.

- The diffusion operator for SLS based on GCM can be written as

$$\mathcal{L}_{\text{diff}}^{n+1}\left(\Phi_p\right) = \nabla_c^x \left(\beta_p \nabla_{f,gh}^x \Phi_p^{n+1}\right) + \nabla_c^y \left(\beta_p \nabla_{f,gh}^y \Phi_p^{n+1}\right) \quad , \tag{9.18}$$

with $\nabla_{f,gh}^x$ and $\nabla_{f,gh}^y$ the ghost gradient operators defined such that ghost values are used when the stencil crosses the interface. This gives (here for $\nabla_{f,gh}^x \Phi\Big|_{i-\frac{1}{2},j}$ with $\mathcal{C}_{i-1}$ in the other phase)

$$\nabla_{f,gh}^x \Phi_p\Big|_{i-\frac{1}{2},j} = \frac{\Phi_{p,i,j} - \Phi_{p,i-1,j}^{gh}}{\Delta x} \quad . \tag{9.19}$$

Then the ghost values are defined using the methodology detailed in Section 8.3.3.4. For the energy equation, the interface temperature T_Γ is prescribed, while the Robin boundary condition $Y_\Gamma \frac{\dot{m}}{\rho_g \mathcal{D}_v} + Y^{(1)} = \frac{\dot{m}}{\rho_g \mathcal{D}_v}$ is prescribed for the species mass fraction. A table to sum up the different terms depending on the scalar is given in Table 9.1 with $\theta = |x_p - x_\Gamma| / \Delta x$.

Φ_p	T_l	T_g	Y
β_p	$\frac{k_l}{\rho_l c_{p,l}}$	$\frac{k_g}{\rho_g c_{p,g}}$	$\mathcal{D}_v$
$F_\Gamma^{(\Phi_p)}$	$\frac{k_l}{\rho_l c_{p,l}} \frac{T_l^{n+1} - T_\Gamma^n}{\|\mathbf{x}_l - \mathbf{x}_\Gamma\|_2}$	$\frac{k_g}{\rho_g c_{p,g}} \frac{T_g^{n+1} - T_\Gamma^n}{\|\mathbf{x}_g - \mathbf{x}_\Gamma\|_2}$	$\frac{\dot{m}^n}{\rho_g}\left(1 - Y^{n+1}\right)$
Φ_p^{gh}	$\frac{T_\Gamma^n + (\theta-1) T_l^{n+1}}{\theta}$	$\frac{T_\Gamma^n + (\theta-1) T_g^{n+1}}{\theta}$	Eq. (B.3) on Y^n

Table 9.1: *Summary of term definitions for the three scalar equations*

Note that only the Robin condition applied on Y using the general normal direction discretization of Section 8.3.2 is expressed explicitly while all other boundary conditions includes quantities at t^{n+1}. In the above two-scalar procedure, the value of T_Γ^n and $\dot{m}^n$ are required and need a closure at the interface.

9.1.4 Phase change procedure

9.1.4.1 Evaporation rate

The most important quantity in a phase change solver is the evaporation rate $\dot{m}$. From Section 7.1.1, it is clear that using Eq. (1.53) is the most promising approach to handle general vaporization problems with arbitrarily high evaporation rate as it is always definedd even when Y_Γ is close to one.
In this work, the evaporation rate is then defined by

$$\dot{m} = \frac{k_g T_{\Gamma,g}^{(1)} - k_l T_{\Gamma,l}^{(1)}}{\mathscr{L}_{vap} - [c_p]_\Gamma (T_\Gamma - T_{sat})} \quad . \tag{9.20}$$

The normal gradients of temperature $T_{\Gamma,l}^{(1)}$ and $T_{\Gamma,g}^{(1)}$ are computed using LE or QE extrapolation. The numerical test cases presented in Section 9.2 are dedicated to compare them for evaporation simulations.
Both procedures need the extrapolation of T_l^n and T_g^n in the other phase following the procedure of Appendix B.1.1. It has been explained in Section 9.1.2 that $\dot{m}$ is needed in a narrow band around the interface to properly define the interface velocity $\mathbf{u}_\Gamma$ from Eq. (9.8). Therefore, a constant extension of $\dot{m}$ from interface cells Ω_Γ to a narrow band around the interface is done using the strategy presented in Appendix B.2.5 with $m = 1$. After this step, $\dot{m}$ is completely defined in the narrow band. The details on the discretization of these PDE extrapolation can be found in Appendix B.3.

9.1.4.2 Interface temperature and species mass fraction

In the solver, an iterative procedure is employed to obtain simultaneously T_Γ and Y_Γ. This is done by using the procedure of [159] and [205] for VOF and SLS respectively which allows a robust manner to reconstruct the quantities at the interface and does not require more than 5 iterations to converge using a regula falsi method.
The two methods only differ in the manner to evaluate the gradients to close the system. They both rely on the discretization of $F_\Gamma^{(\Phi_p)}$ reported in Table 9.1 for a Dirichlet boundary condition.

9.1.4.3 Coupling with energy and species equations

The overall phase change procedure needs the combination of three important features:

1. The computation of $\dot{m}$;
2. The computation of T_Γ and Y_Γ;
3. The boundary condition at the interface for T_l, T_g and Y;

The choice of applying a Robin boundary condition on the species instead of a classic Dirichlet boundary condition Y_Γ is not innocuous. The reason for that is the tight coupling between the scalars and the computation of $\dot{m}$, T_Γ and Y_Γ. This point is discussed in details on a planar Stefan flow problem in Section 9.2.1.3 where multiple combinations are tested to bring out the complexity of choosing a combination which provides a limited coupling

between the three features. However, the combination given here (and used in the literature [183; 159]) provides a viable phase change procedure for general vaporization problems. Here, the coupling of $\dot{m}$ with pressure and interface regression is not discussed as it is expected to be a fairly low coupling compared to energy and species equations.

9.1.5 Summary of the phase change solver procedure

A summary of the adaptations introduced to use VOF and SLS methods in the same low Mach solver is provided in Table 9.2 and the full algorithm for a time step is given:

1. Compute ρ_f^n, α_l^n and α_g^n from c^n;
2. Compute $\mathbf{u}_\Gamma^n$ following the procedure of Section 9.1.2;
3. Advance interface following one of the algorithm of Section 4.1 to obtain c^{n+1};
4. Compute $\mathbf{n}^{n+1}$, κ_f^{n+1}, $H_\Gamma^{0,n+1}$, δ_Γ^{n+1}, μ^{n+1}, ρ_f^{n+1}, α_l^{n+1} and α_g^{n+1} from c^{n+1};
5. Set $\mathbf{u}^k = \mathbf{u}^n$ and loop in a RK2-SSP time integration from $k = 1$ to $k = 3$
   (i) Extrapolate constantly $\mathbf{u}_l^k$ and $\mathbf{u}_g^k$ from $\mathbf{u}^k$
   (ii) Advance α_l^n and α_g^n to α_l^* and α_g^*
   (iii) Advance T_l^n, T_g^n and Y^n to T_l^{k+1}, T_g^{k+1} and Y^{k+1}
   (iv) Extrapolate T_l^{k+1}, T_g^{k+1} and Y^{k+1} following procedure in Section 9.1.4.1
   (v) Update $\dot{m}^{k+1}$, T_Γ^{k+1} and Y_Γ^{k+1} from T_l^{k+1}, T_g^{k+1} and Y^{k+1};
   (vi) Advance $\mathbf{u}^n$ and ρ_f^n to ρ_f^* and $\mathbf{u}^*$ with the consistent mass and momentum scheme of Section 6.1.1 and fluxes computed using $\mathbf{u}^k$;
   (vii) Solve the Pressure from Eq. (9.1);
   (viii) Correct the velocity to obtain $\mathbf{u}^{k+1}$ with Eq. (6.17);

The timestep has a stability constraint based on the velocity, the surface tension and the viscosity :

$$\Delta t < \min\left(\frac{\Delta x}{2\|\mathbf{u}\|}, \sqrt{\frac{\Delta x^3(\rho_l+\rho_g)}{(2\pi)^3\sigma}}, \frac{\Delta x^2}{4\max(\nu_l,\nu_g)}\right) \quad . \tag{9.21}$$

In most of the application shown hereafter, the surface tension restriction is dominant. The absence of stability constraint related to energy and species diffusivity is due to the implicit scheme employed in the unified framework.

method	VOF	SLS
$\mathbf{n}$	VOF-P	SLS-D
κ	HF	FD
κ_f	Eq. (5.74)	Eq. (5.70)
H_Γ^ϵ, α_p	f	Eq. (9.15)
θ_f	PLIC	Eq. (6.11)
δ_Γ	PLIC	GFM
$\dot{m}$	LSL	GCM
IBM	CCM	GCM

Table 9.2: *Summary of computation choices for VOF and SLS*

9.2 Numerical results

In this section, the goal is to investigate the use of linear and quadratic extrapolations for gradient reconstructions at the interface and the combination choices in the phase-change procedure presented in Section 9.1.4.3. The relevance of quadratic extrapolation needs to be clarified as it implies additional PDEs to solve and then more computational time.
For this purpose, planar Stefan flow and spherical Stefan flow problems are investigated. The planar solutions allow to consider 1D simulations where the interface topology does not affect the results. Then, the quantity reconstruction at the interface are not prompt to geometric errors which can be prevalent as demonstrated in Section 8.2.1.1. In the 3D test case, topology errors are also included and allows to conclude on the extrapolation choice for the relevant applications implying topology deformations.
In Section 8.2.4, the methods using the functional fitting either based on linear or quadratic extrapolations SL-LE and LSL-QE have been retained for VOF whereas the ghost-cell method with or without interpolation of the gradients based on either linear or quadratic extrapolations NI-LE and LI-QE have been retained for SLS. These methods are investigated for all test cases and are referred as VOF-LE, VOF-QE, SLS-LE and SLS-QE respectively.

9.2.1 Planar Stefan problems

The planar Stefan flow problems considered here are successively introduced with different levels of coupling between mass, momentum, energy and species given in Table 9.3. This allows a step-by-step validation of the solver along with the impact of the solver choices for different degree of complexity in the test case investigated.

Case	Frank sphere	Planar boiling	Planar evaporation
c	**on**	**on**	**on**
$\dot{m}$	**on**	**on**	**on**
T_l	**on**	off	**on**
T_g	off	**on**	**on**
u	off	**on**	**on**
Y	off	off	**on**
T_Γ, Y_Γ	off	off	**on**

Table 9.3: *Activated equations of the solver for the planar Stefan problems*

All the presented planar Stefan flow problems consist of the same domain of length L, with wall boundaries at the left T_w, Y_w and outflow boundaries at the right T_∞, Y_∞ as illustrated in Fig. 9.1. The gas phase is located between the wall and the interface while the liquid domain is supposed infinite starting from the interface position.
Phase change occurs because of the jump in energy flux or species flux at the interface and creates the motion of the interface through [67; 76; 7; 159]

$$x_\Gamma = 2\gamma\sqrt{\lambda t} \quad , \tag{9.22}$$

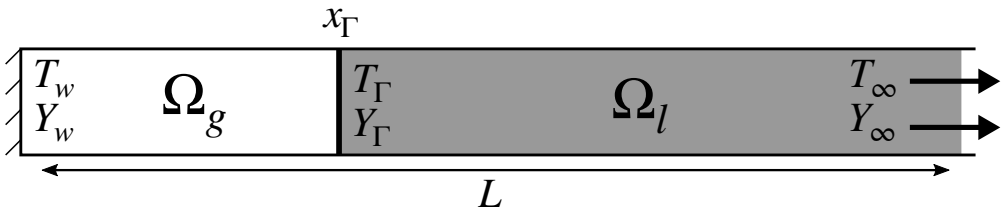


Figure 9.1: *Illustration of the planar Stefan problem*

with γ a diffusion layer and the thermal diffusivity such that $\lambda = k/(\rho c_p)$.
The interface velocity is then deduced from Eq. (9.22):

$$u_\Gamma = \gamma\sqrt{\frac{\lambda}{t}} \quad . \tag{9.23}$$

By using the fact that $u_g = 0$, the liquid velocity can be computed analytically as

$$u_l = \gamma\sqrt{\frac{\lambda}{t}}\left(1 - \frac{\rho_g}{\rho_l}\right) \quad . \tag{9.24}$$

The analytical solution of temperature and species depends on the Stefan problem and is given in the corresponding sections. The fluid properties and physical set-up used in these problems are given in Table 9.4 and Table 9.5 respectively.

Case	Frank sphere		Planar boiling		Planar evaporation	
Phase	Liquid	Gas	Liquid	Gas	Liquid	Gas
ρ (kg/m^3)	1	1	1000	1	1000	1
μ (Pa $\cdot$ s)	1	1	$1 \cdot 10^{-2}$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-5}$
k (W/m/K)	1	1	$1 \cdot 10^{-1}$	$1 \cdot 10^{-2}$	$1 \cdot 10^{-1}$	$1 \cdot 10^{-2}$
c_p (J/kg/K)	1	1	1000	1000	1000	1000
λ (m^2/s)	1	1	$1 \cdot 10^{-7}$	$1 \cdot 10^{-5}$	$1 \cdot 10^{-7}$	$1 \cdot 10^{-5}$
M (kg/mol)	–	–	–	–	–	0.018
$\mathscr{L}_{vap}$ (J/kg)	1	–	$1 \cdot 10^{6}$	–	$1 \cdot 10^{6}$	–
$\mathcal{D}_v$ (m^2/s)	–	–	–	–	–	$1 \cdot 10^{-5}$
T_{sat} (K)	0	–	373.15	–	373.15	–
σ (N/m)	1	–	0.01	–	0.01	–

Table 9.4: *Physical properties of fluids considered in the planar Stefan problems*

In the following, metrics are defined to quantify the errors associated to the 1D test cases. The $L_1(\Phi_p)$ and $L_\infty(\Phi_p)$ errors of the phase scalars with $p = l, g$ are defined as

$$L_1\left(\Phi_p\right) = \frac{\frac{1}{N_p}\sum_{i=1}^{N_p}\left|\Phi_{p,i,exact} - \Phi_{p,i,num}\right|}{\Phi_\infty} \quad , \tag{9.25}$$

$$L_\infty\left(\Phi_p\right) = \max_i \frac{\left|\Phi_{p,i,exact} - \Phi_{p,i,num}\right|}{\Phi_\infty} \quad . \tag{9.26}$$

Case	Frank sphere	Planar boiling	Planar evaporation
L (m)	10	$1 \cdot 10^{-3}$	$1 \cdot 10^{-3}$
t_0 (s)	1	0.1	$1 \cdot 10^{-2}$
t_f (s)	1.5	0.2	0.1
x_Γ (m)	0.86	$1.4119 \cdot 10^{-4}$	$7.3204 \cdot 10^{-5}$
T_Γ (K)	0	373.15	296.12
T_∞ (K)	0.5	373.15	323.15
T_w (K)	0	383.15	296.12
Y_Γ	–	–	0.22106
Y_w	–	–	0.2
Y_∞	–	–	1
γ	0.43	0.0706	0.11575

Table 9.5: *Physical set-up for the planar Stefan problems*

N_p corresponds to the number of cells belonging to the phase p. $\Phi_{p,i,exact}$ is provided by the analytical solution while $\Phi_{p,i,num}$ is the numerical value provided by the solver at the end of the simulation.
The mass error can be expressed in term of the error in the interface position $E(x_\Gamma)$ as

$$E(x_\Gamma) = \frac{|x_{\Gamma,exact} - x_{\Gamma,num}|}{L} \quad , \tag{9.27}$$

with L the domain length $x_{\Gamma,exact}$ the exact interface position and $x_{\Gamma,num}$ the numerical value at the end of the simulation.
Finally, the phase velocity u_p with $p = l, g, \Gamma$ error can be expressed as

$$E(u_p) = \frac{\max\limits_i |u_{p,exact} - u_{p,i,num}|}{u_{p,exact}} \quad , \tag{9.28}$$

with $u_{p,exact}$ the exact phase velocity and $u_{p,i,num}$ the numerical value in $\mathcal{C}_i$ at the end of the simulation.
It is very important that the different quantities of the physical set-up given Table 9.5 are taken with a large number of significant digits to retrieve convergent behaviours. In fact, the errors are already very low in 1D test cases and then become very sensitive to initialization errors.

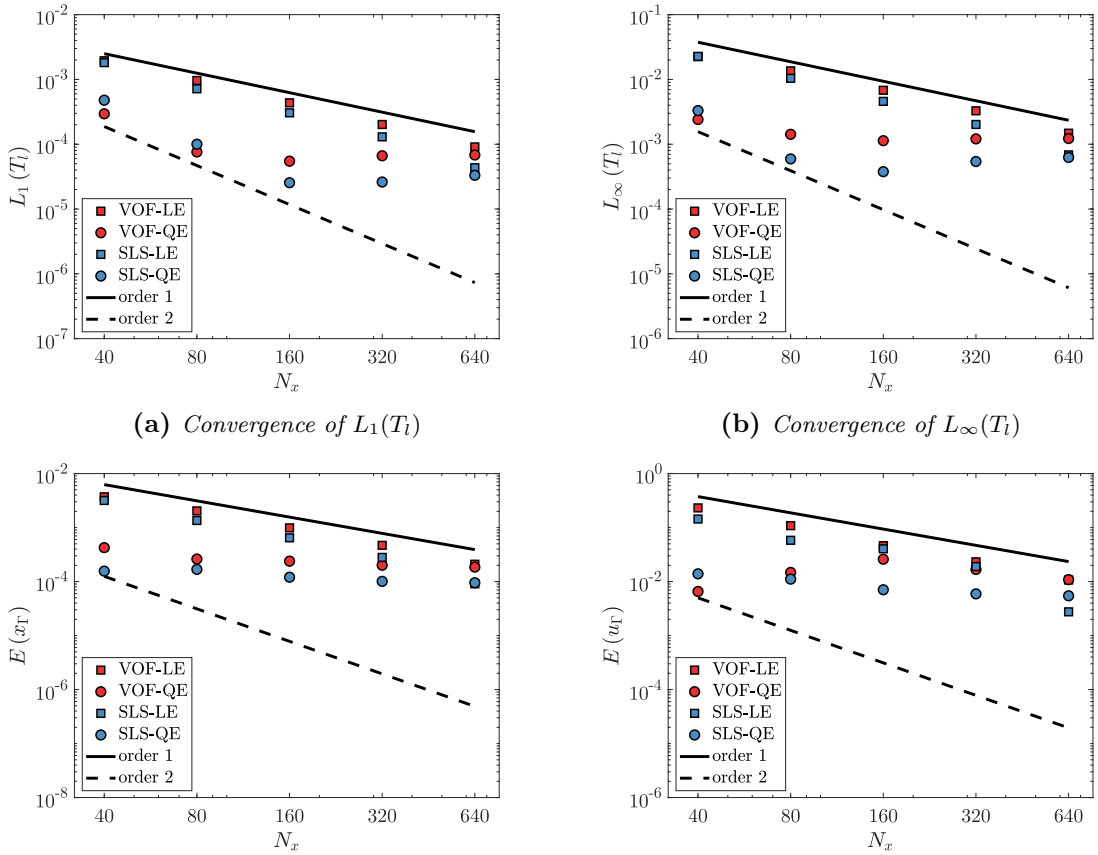
9.2.1.1 Planar Frank sphere

The first test case aims to evaluate the accuracy of both $\dot{m}$ and the diffusion problem with a moving boundary condition. The planar Frank Sphere is a class of problem where two fluids of same properties (see Table 9.4) are separated by an interface. It has been used in [67] to assess a fourth order accurate scheme for the heat equation. In this mathematical problem, the momentum equation is not relevant as $[\rho]_\Gamma = 0$, the velocity is set to zero and does not evolve with time. Moreover, the species mass fraction equation is not considered

neither because both fluids have the same properties, thus $Y = 1$ in the domain. Then the temperature at the interface is at saturation and is fixed to $T_\Gamma = T_{sat}$ given in Table 9.4. As the wall temperature T_w is also set to T_{sat}, the gas temperature is not solved neither and is simply set to $T_g = T_{sat}$. The evolution of liquid temperature is then given by the expression

$$T_l = T_w + \frac{T_\infty - T_\Gamma}{\operatorname{erfc}(\gamma)} \left(\operatorname{erfc}(\gamma) - \operatorname{erfc}\left(\frac{x}{2\sqrt{\lambda_l t}} \right) \right) \quad , \tag{9.29}$$

with the values of T_w, T_∞ and T_Γ given in Table 9.5 and λ_l given in Table 9.4. Therefore, phase change occurs because of the jump in energy flux at the interface and is driven by $\left.\frac{\partial T_l}{\partial x}\right|_\Gamma$. The simulation starts with the analytical temperature of T_l at $t_0 = 1$ s with the initial position of the interface $x_\Gamma = 0.86$ m and stops at $t_f = 1.5$ s.



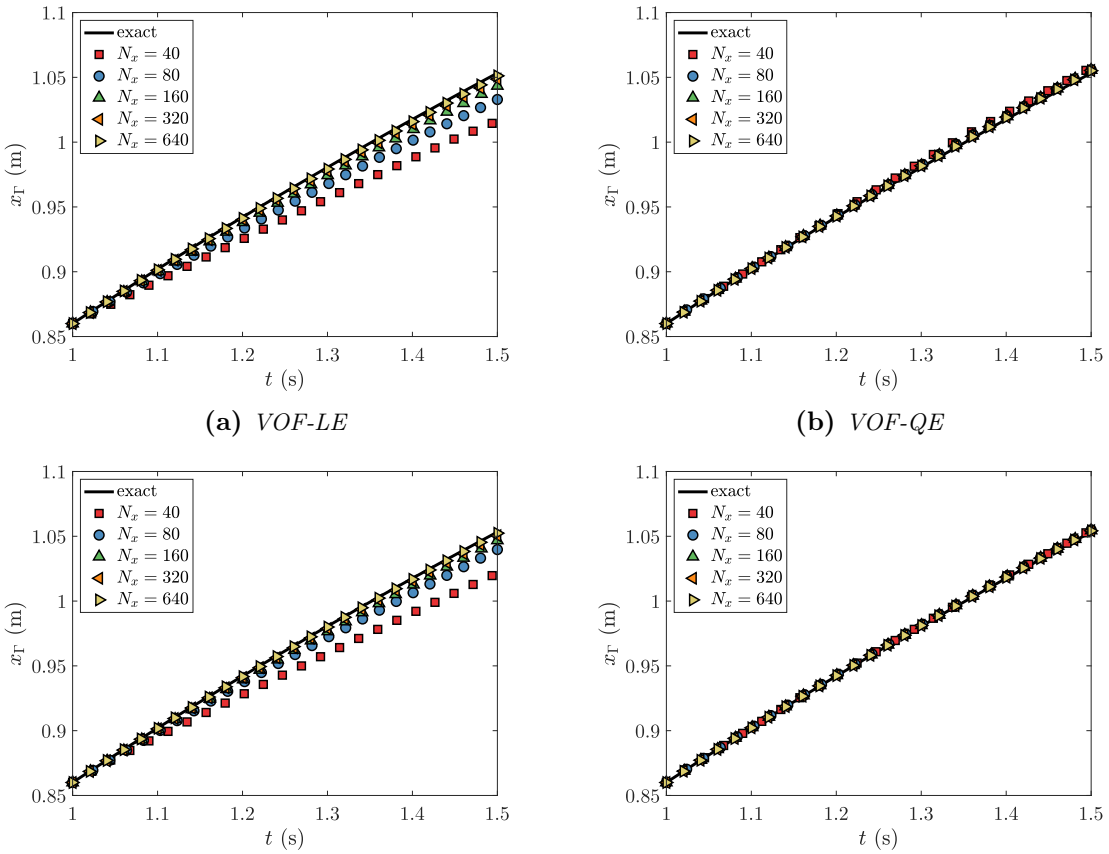
(a) *Convergence of* $L_1(T_l)$ **(b)** *Convergence of* $L_\infty(T_l)$

(c) *Convergence of* $E(x_\Gamma)$ **(d)** *Convergence of* $E(u_\Gamma)$

Figure 9.2: *Mesh convergence of errors for the planar Frank sphere problem*

In Fig. 9.2, the errors associated to the planar Frank sphere problem are provided. First, $L_1(T_l)$ and $L_\infty(T_l)$ exhibits a second-order trend for the coarse meshes in Fig. 9.2a and Fig. 9.2b while it saturates from $N_x = 160$ when using quadratic extrapolations (VOF-QE

and SLS-QE). In fact, using a quadratic extrapolation on a second-order field at most does not lead to second-order accuracy as illustrated in Section 8.2.4 which could explain this saturation. On the other hand, the gradient reconstructions based on linear extrapolation (VOF-LE and SLS-LE) converge between first and second-order. It can be observed that no convergence is retrieved for $E(x_\Gamma)$ and $E(u_\Gamma)$ in Fig. 9.2c and Fig. 9.2d when using quadratic extrapolations. This lack of convergence can be explained by looking at the interface position evolution with time represented in Fig. 9.3. In fact, the first mesh N_x already provide a position evolution which is very close to the exact solution. Therefore, the error related to linear extrapolation is predominant for the LE-based cases which allows the first-order convergence. For the quadratic extrapolation, as the error related to gradient reconstruction is drastically reduced, another source of error could become predominant with mesh resolution leading to the non-convergent behaviour. This source of error could come from the boundary condition applied to liquid temperature which is not at infinity or the time integration.



(a) *VOF-LE* **(b)** *VOF-QE*

(c) *SLS-LE* **(d)** *SLS-QE*

Figure 9.3: *Interface position evolution* $\mathbf{x}_\Gamma(t)$ *for the planar Frank sphere problem*

Overall, even if the QE-based gradient reconstructions do not converge in the high resolution

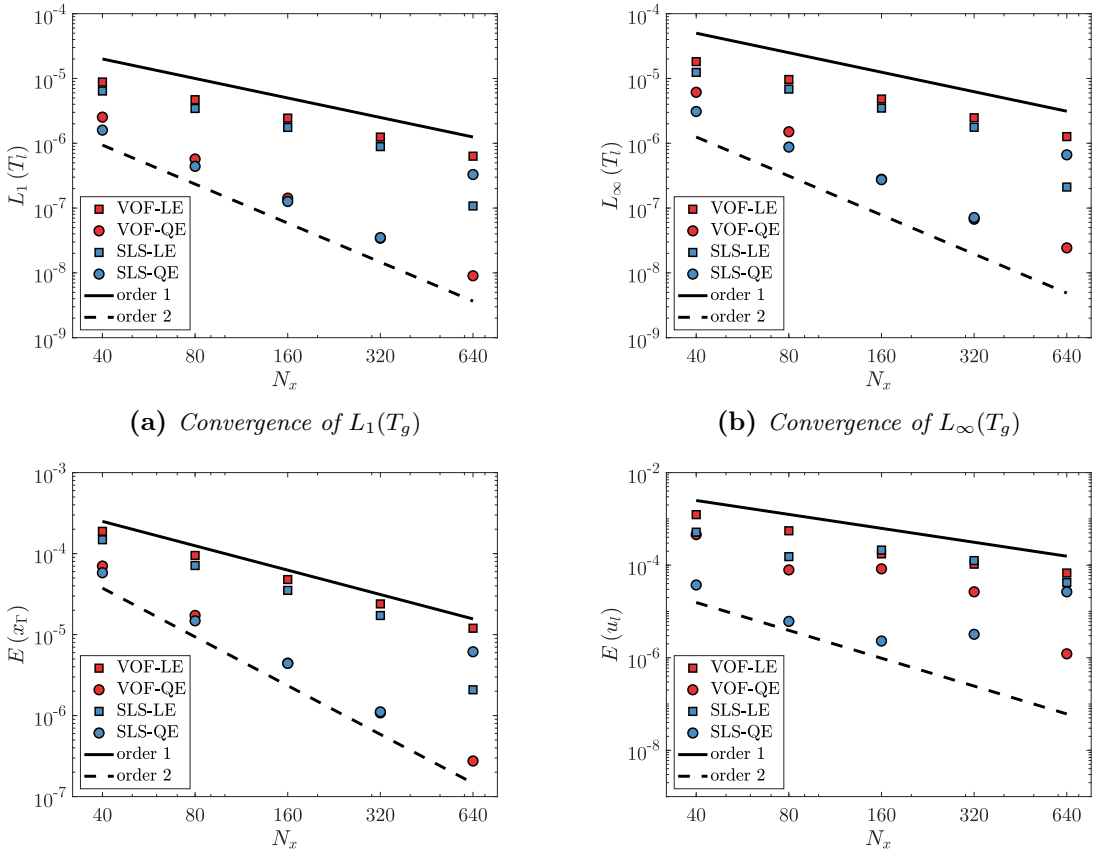
limit, it provides more accurate quantities compared to LE-based gradient reconstructions for low resolutions. Also, SLS is globally more accurate than VOF for all metrics.
From these observations, the choice of extrapolation is not clear, on the one hand, linear extrapolation provides a convergent behaviour but on the other hand, quadratic extrapolation is more accurate for all meshes considered here. However, for a test case with more source of errors as the planar boiling, it is not clear if the quadratic extrapolation still provides more accurate results. At this stage, no conclusion on the best choice can be made and both extrapolations are still compared in the next study about planar boiling.

9.2.1.2 Planar boiling

A step further to general vaporization is to consider a physical problem with a density jump between both phases. Then, the momentum equation is solved with a velocity jump caused by phase change at the interface. The planar boiling problem was first considered in [76] and investigated in [7; 159] where liquid ans gas are considered to be the fluid and its vapour with fluid properties of Table 9.4. Then, the interface is at the saturation conditions of the pure liquid and the species equation is not relevant to solve. The interface temperature is fixed to $T_\Gamma = T_{sat}$. The temperature at infinity is set such that $T_\infty = T_{sat}$, therefore, the liquid temperature is set to $T_l = T_{sat}$ and does not require to be solved in opposition to the Frank sphere problem. The evolution of gas temperature is then given by the expression

$$T_g = T_w + \frac{T_\Gamma - T_w}{\mathrm{erf}\,(\gamma)}\,\mathrm{erf}\left(\frac{x}{2\sqrt{\lambda_g t}}\right) \quad , \tag{9.30}$$

with the values of T_w and T_Γ given in Table 9.5 and λ_g given in Table 9.4. Phase change still occurs because of the jump in energy flux at the interface and is driven by the evaluation of $\frac{\partial T_g}{\partial x}\Big|_\Gamma$. The simulation starts with the analytical temperature of T_g at $t_0 = 0.1$ s with the initial position of the interface $x_\Gamma = 1.4119 \cdot 10^{-4}$ m and stops at $t_f = 0.2$ s.
In Fig. 9.4, the errors associated to the planar boiling problem are provided. As opposed to the Frank sphere problem, a clear second-order convergence is observed in Fig. 9.4a , Fig. 9.4b and Fig. 9.2c for $L_1(T_l)$, $L_\infty(T_l)$ and $E(x_\Gamma)$ when using quadratic extrapolations unless for the $N_x = 640$ simulation using SLS-QE. For the linear extrapolation, the expected first-order convergence is observed. The order of convergence is not clear for $E(u_l)$ in Fig. 9.4d even if a second-order tendency can also been observed for quadratic extrapolations whereas it is first-order for linear extrapolations.
For that test case, the use of quadratic extrapolation leads to a serious improvement in accuracy. It is quite surprising as the system of equations induces more sources of error due to the mass and momentum equations and quadratic extrapolations are not expected to improve the order of convergence as demonstrated in Section 8.2.4. Now, it is interesting to pursue the same study for evaporation where all equations and closures have to be explicitly solved.



(a) *Convergence of* $L_1(T_g)$ **(b)** *Convergence of* $L_\infty(T_g)$

(c) *Convergence of* $E(x_\Gamma)$ **(d)** *Convergence of* $E(u_l)$

Figure 9.4: *Mesh convergence of errors for the planar boiling problem*

9.2.1.3 Planar evaporation

Finally, the general evaporation problem of [159] is presented where the gas part of the domain is a mix of inert gas and liquid vapour. It means that the species mass fraction equation evolves in time and needs to be solved in the gas. The analytical solution for species mass fraction evolution is

$$Y = Y_w + \frac{Y_\Gamma - Y_w}{\text{erf}\,(\gamma)} \,\text{erf}\left(\frac{x}{2\sqrt{\mathcal{D}_v t}}\right) \quad , \tag{9.31}$$

with Y_w and Y_Γ given in Table 9.5 and $\mathcal{D}_v$ given in Table 9.4. Note that γ is the same layer for species and temperature as the Lewis number $Le = \lambda_g/\mathcal{D}_v = 1$ in this case.
In such case, the quantities at the interface are not imposed explicitly and need to be retrieved from the phase-change procedure. The exact value of T_Γ and Y_Γ can be derived

analytically with the additional relation from Eqs. (1.53) and (1.54) as in [159]:

$$T_\Gamma = T_w + \frac{\mathscr{L}_{vap}}{c_{p,g}} \sqrt{\frac{\mathcal{D}_v}{\lambda_g}} \frac{Y_\Gamma - Y_w}{Y_\Gamma - 1} \quad , \tag{9.32}$$

and using Clausius-Clayperon to close the system. This leads to $T_\Gamma = 296.12$ K and $Y_\Gamma = 0.22106$.
As T_Γ is computed numerically, it does not match exactly the initial liquid temperature $T_l = T_\infty$. Then the liquid temperature also needs to be solved to take into account the potential derivations of T_Γ from T_∞.
The evolution of gas temperature is still

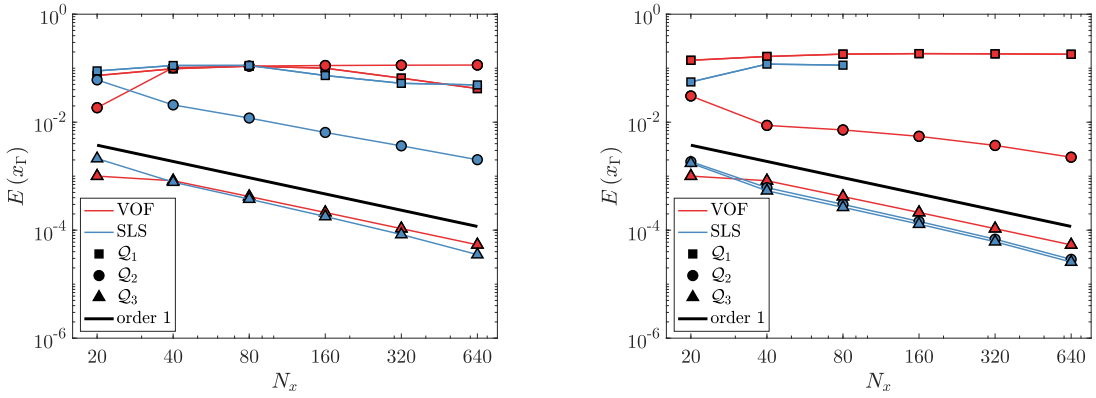
$$T_g = T_w + \frac{T_\Gamma - T_w}{\text{erf}\,(\gamma)} \text{erf}\left(\frac{x}{2\sqrt{\lambda_g t}}\right) \quad , \tag{9.33}$$

with the values of T_w and T_Γ given in Table 9.5 and λ_g given in Table 9.4. Phase change still occurs because of the jump in energy flux at the interface and is driven by the evaluation of $\frac{\partial T_g}{\partial x}\Big|_\Gamma$. The simulation starts with the analytical temperature of T_g at $t_0 = 0.01$ s with the initial position of the interface $x_\Gamma = 7.3204 \cdot 10^{-5}$ m and stops at $t_f = 0.1$ s.

Full phase change procedure study As the interface quantities T_Γ and Y_Γ are not imposed anymore and needs an explicit reconstruction at the interface, this last test case allows to investigate the phase-change procedure. The aim of this study is to bring out the impact of the different combinations of phase change procedure discussed in Section 9.1.4.3. Some key points of the following discussion are already notified in [159].
First, the Dirichlet boundary condition applied to Y is investigated by trying different quantity reconstruction at the interface. Then, the same study is performed with Robin boundary condition applied to Y for comparison.
Different reconstructions of Y_Γ and T_Γ are investigated for this choice of embedded boundary conditions at the interface:

- A first-order extrapolation $\mathcal{E}_1(Y)$ presented in Section 7.1.2 leading to method $\mathcal{Q}_1(Y)$
- A second-order extrapolation $\mathcal{E}_2(Y)$ presented in Section 7.1.2 leading to method $\mathcal{Q}_2(Y)$
- An iterative process based on the general system presented in Section 7.1.1 with the gradient reconstruction of [159] and [205] for VOF and SLS respectively. This leads to method $\mathcal{Q}_3(T_l, T_g, Y)$

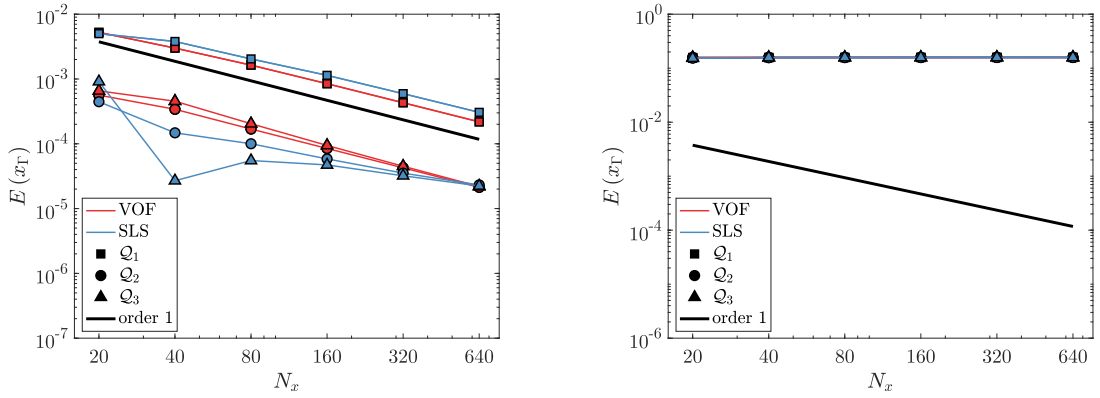
In the results shown here, the evaporation rate is computed either from the temperature $\mathcal{M}_1(T_l, T_g)$ or the species mass fraction $\mathcal{M}_2(Y)$.
All the analysis rely on the interface position error $E(x_\Gamma)$ as it is sufficient to detect inconsistent behaviour of the phase-change procedure.
First, the phase change procedure is designed to impose Dirichlet boundary conditions for all scalars T_l, T_g and Y. The boundary conditions are then direct functions of the quantities at the interface $\mathcal{Q}$, this is expressed as $T_l|_{\partial\Omega}(\mathcal{Q})$, $T_g|_{\partial\Omega}(\mathcal{Q})$ and $Y|_{\partial\Omega}(\mathcal{Q})$.



(a) $\mathcal{M}_1(T_l, T_g)$ **(b)** $\mathcal{M}_2(Y)$

Figure 9.5: *Mesh convergence of* $E(x_\Gamma)$ *for the planar evaporation problem with different* $\mathcal{Q}$ *and* $\mathcal{M}$ *for Dirichlet boundary conditions applied to* Y

In Fig. 9.5, the mesh convergence of $E(x_\Gamma)$ is given. As expected, the reconstruction based on $\mathcal{Q}_1(Y)$ leads to an inconsistent scheme both with $\mathcal{M}_1(T_l, T_g)$ and $\mathcal{M}_2(Y)$ (see Fig. 9.5a Fig. 9.5a respectively). This is due to the strong coupling between $\mathcal{Q}_1(Y)$ and $Y|_{\partial\Omega}(\mathcal{Q})$ as Y_Γ is directly set to the closest point to the interface. Then the diffusion operator of Y is inconsistent in the case where the cell center belongs to the gas phase: in such case, the interface flux is always $F_\Gamma^{(Y)} = 0$.
However, the extrapolation procedure allows to retrieve convergent behaviours. For SLS, using $\mathcal{Q}_2(Y)$ relaxes the coupling but can lead to inconsistent fluxes too in the limit cases where the interface is close to a cell center. For VOF, analysis on $\mathcal{Q}_2(Y)$ is not as simple because of the least square system solved to extrapolated the values, however relaxation of the coupling is observed using $\mathcal{M}_2(Y)$ while it is inconsistent using $\mathcal{M}_1(T_l, T_g)$.
Finally, the use of $\mathcal{Q}_3(T_l, T_g, Y)$ drastically lowers the coupling as the quantities depend on all scalar fields and not only on Y. Then the results are improved and retrieve a convergent behaviour.
The same phase change procedure is used but with a Robin boundary condition for Y with the mesh convergence of $E(x_\Gamma)$ given in Fig. 9.6.
This new choice of boundary condition creates a dependency on the evaporation rate computed with $\mathcal{M}$ instead of the quantities at the interface. The boundary condition applied to Y is then a function of $Y|_{\partial\Omega}(\mathcal{M})$ either computed from $\mathcal{M}_1(T_l, T_g)$ or $\mathcal{M}_2(Y)$. As illustrated in Fig. 9.6a, this modification of the boundary condition allows to be consistent using any of the quantity methodologies for both VOF and SLS if the evaporation rate is retrieve from $\mathcal{M}_1(T_l, T_g)$. This is explained by the low coupling between the boundary condition obtained from the temperature fields and the quantity reconstruction obtained from extrapolation of Y. However, when $\mathcal{M}_2(Y)$ is used instead, a tight coupling is created between the boundary condition applied on Y and the field Y itself. This only leads to inconsistent schemes as shown in Fig. 9.6b.
The above study is summarized in Table 9.6 where all well-suited procedures for phase



(a) $\mathcal{M}_1(T_l, T_g)$ **(b)** $\mathcal{M}_2(Y)$

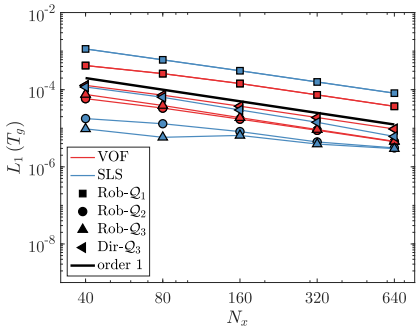
Figure 9.6: *Mesh convergence of* $E(x_\Gamma)$ *for the planar evaporation problem with different* $\mathcal{Q}$ *and* $\mathcal{M}$ *for Robin boundary conditions applied to* Y

change are represented with ✓ while inconsistent procedures are tagged by ✗.

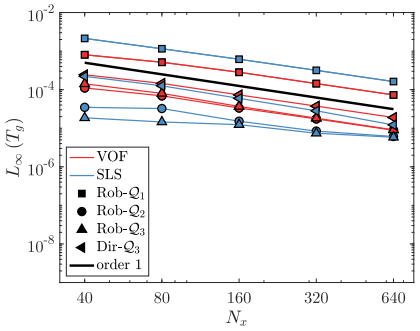
		Dirichlet		Robin	
		$\mathcal{M}_1(T_l, T_g)$	$\mathcal{M}_2(Y)$	$\mathcal{M}_1(T_l, T_g)$	$\mathcal{M}_2(Y)$
VOF	$\mathcal{Q}_1(Y)$	✗	✗	✓	✗
	$\mathcal{Q}_2(Y)$	✗	✓	✓	✗
	$\mathcal{Q}_3(T_l, T_g, Y)$	✓	✓	✓	✗
SLS	$\mathcal{Q}_1(Y)$	✗	✗	✓	✗
	$\mathcal{Q}_2(Y)$	✓	✓	✓	✗
	$\mathcal{Q}_3(T_l, T_g, Y)$	✓	✓	✓	✗

Table 9.6: *Sum-up of well-suited (✓) and inconsistent (✗) procedures for phase change.*

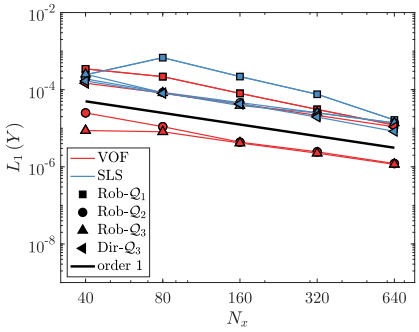
Note that a well-suited method leads to convergent behaviour but with huge disparity in the accuracy. Therefore, they need to be compared to pick the most accurate one. In the present solver, the choice has been to use $\mathcal{M}_1(T_l, T_g)$ by default, because it gives a natural way to handle both boiling and evaporation without problems. Then, some additional phase-change procedures does not need to be considered. The choice has been made to compare four methods for both SLS and VOF: Dirichlet boundaries with $\mathcal{Q}_3(T_l, T_g, Y)$ and Robin boundaries with all $\mathcal{Q}$ methods.
Looking at mesh convergence of the different metrics in Fig. 9.7, it is clear that $\mathcal{Q}_1$ provides the less accurate results. Then, depending of the metrics, Robin boundaries with either $\mathcal{Q}_2$ or $\mathcal{Q}_3$ give the most accurate results for both VOF and SLS. One exception is the liquid temperature in the case of a VOF with Dirichlet boundary conditions: the liquid temperature is almost exact. In fact, the phase change procedure in that case is designed to have discrete consistence between the reconstructed values T_Γ, Y_Γ and the embedded Dirichlet boundaries.
As for the boiling, the convergence of $E(u_l)$ is not clear even if it shows a global first-order



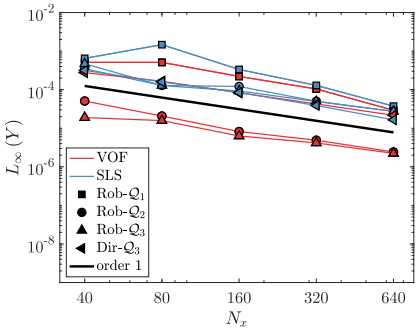
(a) *Convergence of* $L_1(T_g)$



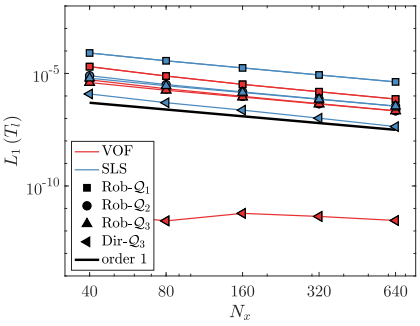
(b) *Convergence of* $L_\infty(T_g)$



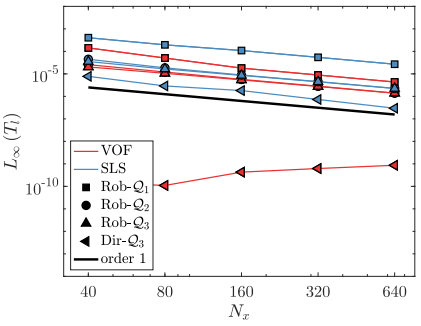
(c) *Convergence of* $L_1(Y)$



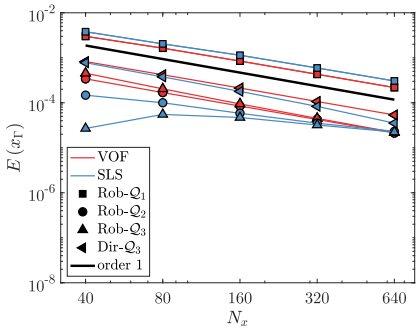
(d) *Convergence of* $L_\infty(Y)$



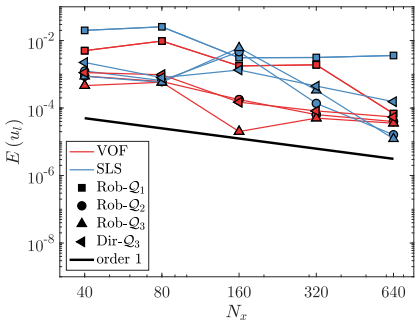
(e) *Convergence of* $L_1(T_l)$



(f) *Convergence of* $L_\infty(T_l)$



(g) *Convergence of* $E(x_\Gamma)$



(h) *Convergence of* $E(u_l)$

Figure 9.7: *Mesh convergence of errors for the planar evaporation problem with different phase-change procedures*

trend.
Finally, regarding implementation and computational efforts, $\mathcal{Q}_2$ is better than $\mathcal{Q}_3$ for a comparable accuracy. From the above considerations, using Robin boundary conditions for species with the $\mathcal{Q}_2$ reconstruction seem to be the best trade off between implementation complexity, efficiency and accuracy.

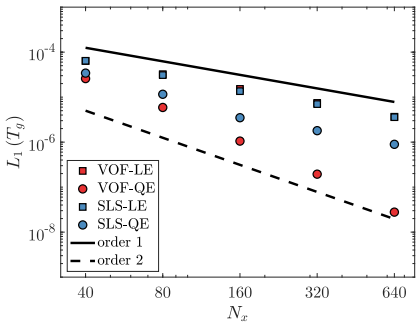
Gradient reconstruction study Now that the phase change procedure as been chosen, the study of Section 9.2.1.1 and Section 9.2.1.2 can be performed to compare linear and quadratic extrapolations.
Again, the use of quadratic extrapolation for the VOF procedure leads to a second-order convergence for all error metrics as shown in Fig. 9.8. However, this is not the case for the SLS which is between first and second-order accuracy. By looking at Figs. 9.8c and 9.8d, the SLS-QE method is less accurate than SLS-LE which can explain the lack of second-order convergence for the other quantities as they are all coupled.
On the other hand, the linear extrapolation leads to an expected first-order convergence of the quantities as already presented in [159].
Overall, the same conclusions can be drawn for the general evaporation test case: using quadratic extrapolation to reconstruct the gradient at the interface drastically improves the results and provides a better convergence rate.

9.2.1.4 Conclusion

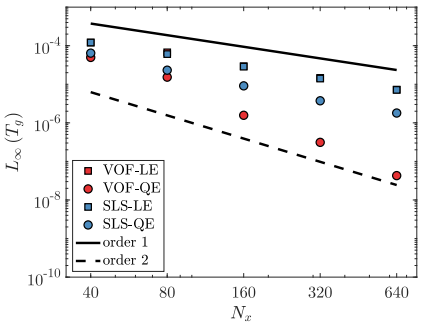
The investigation of 1D test cases raised the key points of a well-suited procedure for phase change with accurate gradient reconstructions:

1. A phase change procedure has to be composed carefully to limit the coupling between the scalars: some combinations have been demonstrated to be inconsistent.
2. The way to compute T_Γ and Y_Γ have an important impact on the evaporation solver accuracy.
3. The gradient reconstruction based on quadratic extrapolations is always more accurate compared to linear extrapolation. While it can suffer from lack of convergence in the high resolution limit for the Frank sphere, it is increasing the convergence rate for boiling and general evaporation which constitutes a serious improvement.

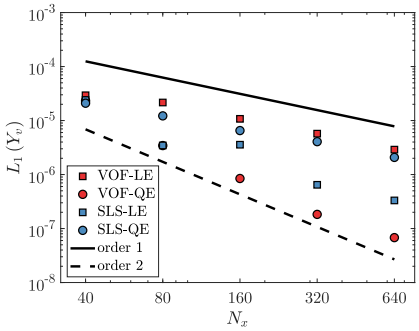
The evaporation solver has shown convergent behaviour for all test cases investigated here demonstrating its feasibility for controlled and accurate 1D phase-change simulations.
Now, a spherical Stefan problem is studied to quantify the multidimensional effects on phase change accuracy. The use of linear and quadratic extrapolations is still investigated as it could lead to different conclusions.
Note that in the multidimensional test cases presented, Dirichlet boundary conditions are used instead of Robin for the species mass fraction as it seemed to provide more robust phase-change simulations.



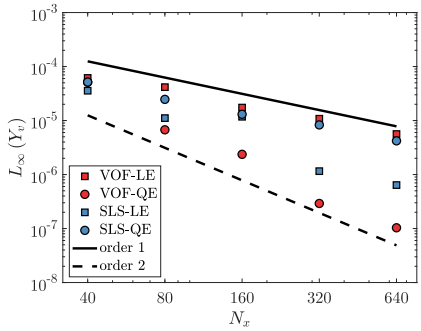
(a) *Convergence of* $L_1(T_g)$



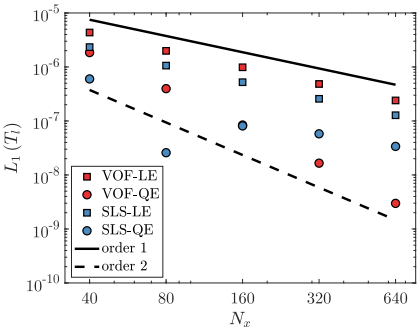
(b) *Convergence of* $L_\infty(T_g)$



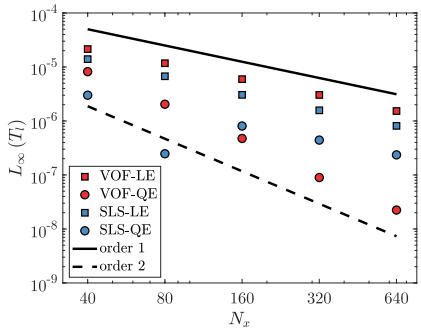
(c) *Convergence of* $L_1(Y)$



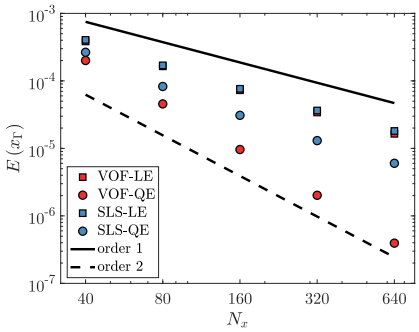
(d) *Convergence of* $L_\infty(Y)$



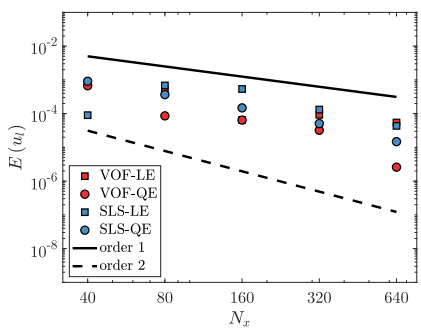
(e) *Convergence of* $L_1(T_l)$



(f) *Convergence of* $L_\infty(T_l)$



(g) *Convergence of* $E(x_\Gamma)$



(h) *Convergence of* $E(u_l)$

Figure 9.8: *Mesh convergence of errors for the planar evaporation problem with different gradient reconstructions*

9.2.2 Static droplet evaporation

The static droplet evaporation is a spherical Stefan problem where topology evaluation takes an important role in the gradient and quantities reconstruction. The test case is also expected to be impacted by the ability of the interface tracking method to regress through transport. In 1D, the transport was almost exact, while in multidimensional test cases the interface capturing method has a huge impact. Here the goal is to assess the VOF and SLS solvers in a general multidimensional evaporation problem with convection effects limited to the Stefan flow induced by the evaporation process.

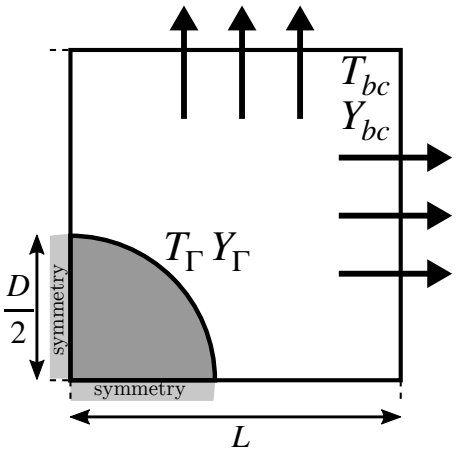


Figure 9.9: *Illustration of the static droplet evaporation*

Case	Stefan spherical	
Phase	Liquid	Gas
ρ (kg/m^3)	700	1
μ (Pa · s)	$3.26 \cdot 10^{-4}$	$1 \cdot 10^{-5}$
k (W/m/K)	$1.61 \cdot 10^{-1}$	$5.2 \cdot 10^{-2}$
c_p (J/kg/K)	2000	1000
M (kg/mol)	0.058	0.029
$\mathscr{L}_{vap}$ (J/kg)	–	$2.3 \cdot 10^6$
D_m (m^2/s)	–	$5.2 \cdot 10^{-5}$
T_{sat} (K)	329	–
σ (N/m)	0	–

Table 9.7: *Physical properties of fluids in the static droplet evaporation*

An acetone droplet of diameter $D = 0.1$ mm is placed in quiescent air and experience evaporation due to the gas conditions at infinity $T_\infty = 700$ K and $Y_\infty = 0$ with the fluid properties reported in Table 9.7. The problem of an isolated droplet evaporating in a gas at rest has been widely investigated in he literature. By assuming a quasi-steady gas phase consisting of the vapour and an inert gas, a pure liquid phase with uniform and stationary temperature, an analytical solution can be derived for temperature and species mass fraction fields along with droplet diameter evolution. The derivation of the solution can be found in [183; 4] and leads to the analytical solutions

$$T_g = T_\infty + \frac{\mathscr{L}_{vap}}{c_{p,g}} \left(1 - (1 - B_T)^{\left(1 - \frac{d}{D_\Gamma}\right)} \right) \tag{9.34}$$

$$Y = 1 + (Y_\infty + 1) \left(1 - (1 + B_M)^{\left(1 - \frac{d}{D_\Gamma}\right)} \right) \quad , \tag{9.35}$$

with $d = (x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2$ the diameter coordinate, D_Γ the droplet diameter, B_T and B_M the Spalding numbers related to heat and mass transfer respectively such that

$$B_T = \frac{c_{p,g} \left(T_\infty - T_\Gamma \right)}{\mathscr{L}_{vap}} \quad , \qquad B_M = \frac{Y_\infty - Y_\Gamma}{Y_\Gamma - 1} \quad . \tag{9.36}$$

The temperature T_l is expected to be equal to the interface temperature $T_l = T_\Gamma$.
As for the planar Stefan flow, the value of T_Γ and Y_Γ can be derived analytically with the additional relation from Eqs. (1.53) and (1.54):

$$T_\Gamma = T_\infty - \frac{\mathscr{L}_{vap}}{c_{p,g}} \left(1 - \left(\frac{Y_\infty - 1}{Y_\Gamma - 1} \right)^{Le} \right) \quad , \tag{9.37}$$

with $Le = \lambda_g / \mathcal{D}_v$ the Lewis number. By using and Clausius-Clayperon to close the system, $T_\Gamma = 294.94$ K and $Y_\Gamma = 0.43993$. The Spalding numbers can be deduced accordingly: $B_T = B_M = 0.7819$.
Finally, the diameter evolution is

$$D_\Gamma = \sqrt{D_0^2 - \frac{4\rho_g D_m}{\rho_l} \ln\left(1 + B_M\right) t} \quad , \tag{9.38}$$

and the evaporation rate

$$\dot{m} = \frac{2\rho_g \mathcal{D}_v \ln\left(1 + B_M\right)}{D_\Gamma} \quad . \tag{9.39}$$

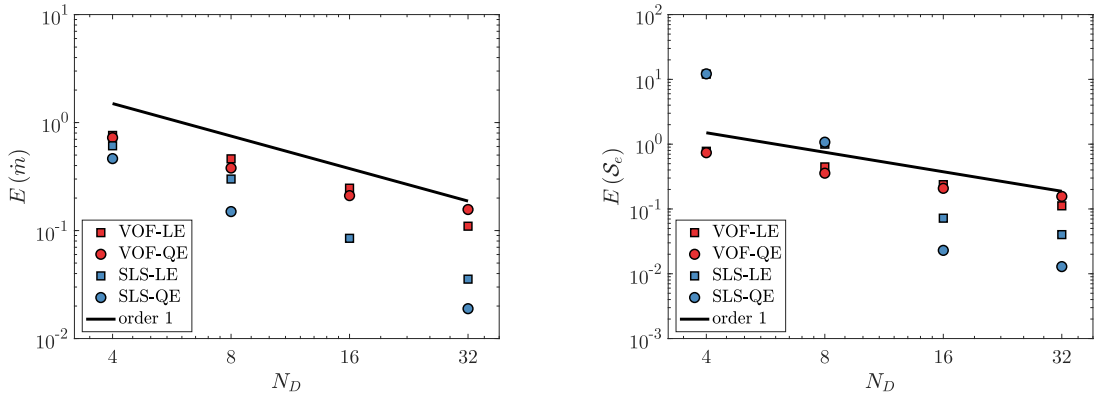
The corresponding evaporation time for this test case with an initial diameter $D_0 = 1 \cdot 10^{-4}$ m is then $\tau_e = 0.029126$ s.
As the test case is performed in 3D on a cartesian grid, only a eighth of the domain is consider by using symmetries as illustrated in Fig. 9.9 to allow a convergence study with reasonable computational time. Moreover, the domain size is reduced to $[0, 4D_0]^3$ such that boundary conditions are not far enough from the droplet to be considered at infinity. Then, the boundary condition for temperature and species mass fraction are imposed by using the exact solution evolving in time $T_{bc}(t, \mathbf{x})$ and $Y_{bc}(t, \mathbf{x})$ as in [183] instead of forcing T_∞ and Y_∞ which would lead to erroneous results. At these boundaries, the flow is allowed to exit freely by imposing zero pressure boundaries.
The simulation is performed during $t_f / \tau_e = 2.5 \times 10^{-3}$ for four different meshes leading to $N_D = 4$ to $N_D = 32$ cells in the diameter for the different methods studied in the planar Stefan flow problem.
The error metrics used here are based on the slope of the d^2 law $\mathcal{S}_e = 1/\tau_e$ and the error on $\dot{m}$ defined as

$$E\left(\mathcal{S}_e\right) = \left| 1 - \frac{1 - \left(\frac{D_{num}}{D_0} \right)^2}{\mathcal{S}_e t_f} \right| \tag{9.40}$$

$$E\left(\dot{m}\right) = \frac{|\dot{m}_{exact} - \dot{m}_{num}|}{\dot{m}_{exact}} \tag{9.41}$$

with D_{num} and $\dot{m}_{num}$ the diameter obtained numerically at t_f with the simulation and $\dot{m}_{exact}$ the exact evaporation rate at t_f.
As the study aims to quantify the accuracy of the phase-change procedure in a multi-dimensional configuration, the potential perturbations caused by the spurious behaviour of

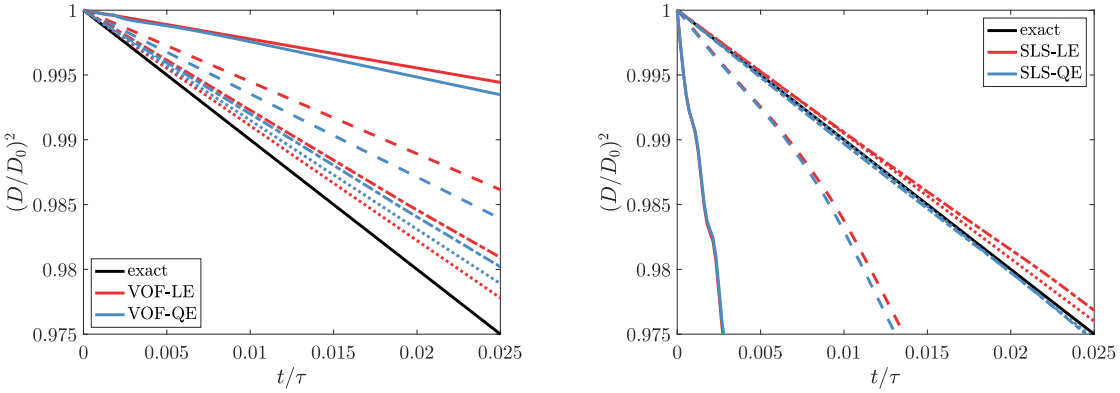
the liquid velocity close to the interface is mitigated by imposing $\kappa = 0$ and $\mathbf{u}_\Gamma = -\dot{m}/\rho_l \mathbf{n}_\Gamma$. However, the true value of $\mathbf{u}_l$ is still used in the transport of T_l. This simplification alleviate potential numerical issues arising from errors in the $\mathbf{u}_l$ reconstruction.



(a) *Convergence of* $E(\dot{m})$ **(b)** *Convergence of* $E(\mathcal{S}_e)$

Figure 9.10: *Mesh convergence of* $E(\dot{m})$ *and* $E(\mathcal{S}_e)$ *for the static droplet evaporation with different extrapolations*

The convergence of $E(\dot{m})$ and $E(\mathcal{S}_e)$ are given in Fig. 9.10. The VOF-LE method converge to first-order for both metrics while VOF-QE loses accuracy for the finest grid $N_D = 32$. Otherwise, VOF-QE leads to a slight improvement in accuracy for the coarse meshes which is not important compared to the gain in the simulation of planar Stefan problems of Section 9.2.1.3. Then multidimensional effects are predominant for the VOF methodology and quadratic extrapolations do not seem to substantially improve results. This observation can explain the choice of linear extrapolation in the work of [159].
For SLS, Fig. 9.10a exhibits higher accuracy compared to VOF. Again, the conclusions are different from the planar test cases where VOF showed the most accurate results. This gives insight in the ability of Level Set to build more accurate reconstructions in multidimensional test cases as it was already the case for the normal and curvature. Moreover, the use of quadratic extrapolations lead to a significant gain in accuracy compared to the linear extrapolations as already observed in [183] using the same methodology.
However, the first two points in Fig. 9.10b exhibits errors higher than 100% of the expected slope $\mathcal{S}_e$ which can be considered as inconsistent solutions. This can be explained by looking at the temporal evolution of the d^2 law in Fig. 9.11. The slopes are always under-predicted for VOF as shown in Fig. 9.11a while in SLS the slope is drastically over-predicted for $N_D = 4$ and $N_D = 8$ in Fig. 9.11b. Then SLS approach is unable to predict the regression correctly for low resolutions. This is explained by the reinitialization step which implies a mass loss which is more important than the regression due to phase change. For completeness, a plot without evaporation has been added in Fig. 9.12 to show mass conservation convergence of the SLS method. It is obvious that the test cases with $N/D = 4$ and 8 cells in the diameter have a regression driven by the reinitialization. However, for better resolutions, the slope is very close to the expected regression when the reinitialization does



(a) *VOF* **(b)** *SLS*

Figure 9.11: *d2-law for VOF and SLS methods for* $N_D = 4$ (——), $N_D = 8$ (- - -), $N_D = 16$ (-···-) *and* $N_D = 32$ (·····).

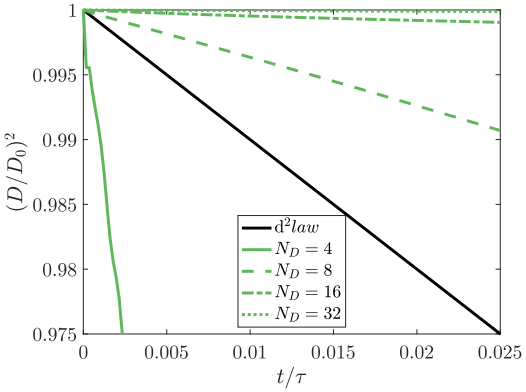


Figure 9.12: *Effect of reinitialization for* $N_D = 4$ (——), $N_D = 8$ (- - -), $N_D = 16$ (-···-) *and* $N_D = 32$ (·····).

not lead to significant mass loss.

To conclude, the multidimensional effects are more visible for VOF than SLS in term of accuracy of $\dot{m}$. This fact was well-known for the reconstruction of $\mathbf{n}$ and κ whereas it was not clear for $\dot{m}$ without this type of comparisons. However, the mass loss induced by reinitialization completely jeopardizes a good prediction of phase change through evaporation when using low resolutions while VOF can still produce reasonable results even if the error is still very large. From all these considerations, the quadratic extrapolations will be used in the application cases as it gives the most accurate predictions of phase change for the range of resolutions used here.

9.3 Convected droplet

Now that canonical test cases of evaporation processes have been explored in 1D and 3D, a 2D application is presented implying convection effects. A similar test case has already been presented in [29], where a water droplet with $D_0 = 5 \cdot 10^{-4}$ m is placed at the top $(2.5D_0, 19D_0)$ of a domain $[0, 5D_0] \times [0, 20D_0]$ with an initial velocity of 1 m/s. The setup is represented in Fig. 9.13 with the associated fluid properties for water and air reported in Table 9.8.

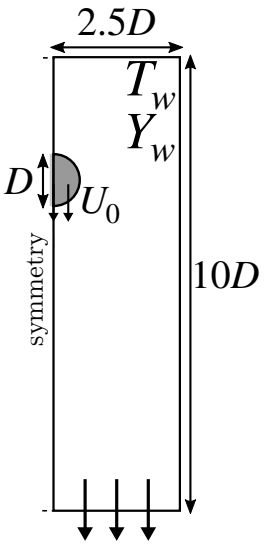


Figure 9.13: *Configuration of the static droplet evaporation*

Case	Stefan spherical	
Phase	Liquid	Gas
ρ (kg/m^3)	1000	1.226
μ (Pa · s) $1.137 \cdot 10^{-2}$	$1.78 \cdot 10^{-5}$	
k (W/m/K)	$6 \cdot 10^{-1}$	$4.6 \cdot 10^{-2}$
c_p (J/kg/K)	1000	4180
M (kg/mol)	0.029	0.018
$\mathscr{L}_{vap}$ (J/kg)	$2.3 \cdot 10^6$	–
D_m (m^2/s)	–	$2 \cdot 10^{-5}$
T_{sat} (K)	373.15	–
σ (N/m)	0.0728	–

Table 9.8: *Physical properties of fluids in the static droplet evaporation*

The left and top boundary conditions are taken as a slipping wall and noslip wall respectively with $T_w = 873$ K and $Y_w = 0$ while the lower boundary is a free outlet. The right boundary is a symmetry to reduce the computational time.
The initial solution is a uniform temperature in the liquid $T_l = 350$ K while T_g and Y are initialized by imposing the spherical Stefan flow solution presented in Section 9.2.2 adapted to the boundary conditions of the present problem. This initial solution is not totally consistent with the physical solution of a cylindrical static droplet evaporating but it provides a decent initial solution for starting the simulation.
Hence, this last case shows the capability of the solver to handle convection effects leading to non-uniform vaporization process and more interactions with the flow. The test case is performed on a low, medium and high resolution mesh corresponding to $N_D = 10, 20$ and 40 respectively.

The mass, kinetic energy and evaporation rate of the droplet are defined as

$$M_l = \sum_{i=1}^{N_{\Omega_l}} \alpha_{l,i}\rho_l \mathcal{V}_i \tag{9.42}$$

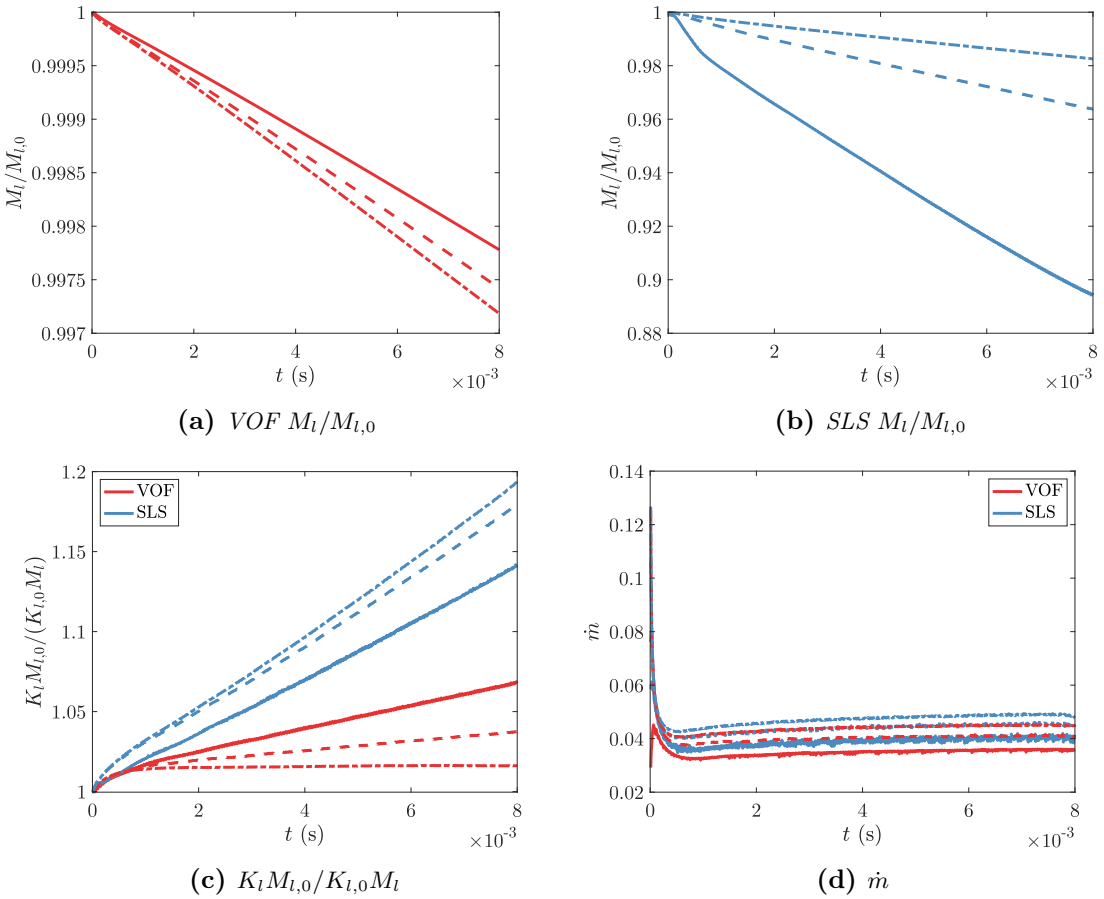
$$K_l = \frac{1}{2}\sum_{i=1}^{N_{\Omega_l}} \alpha_{l,i}\rho_l \|\mathbf{u}_{l,i}\|^2 \mathcal{V}_i \tag{9.43}$$

$$\dot{m} = \frac{\sum_{i=1}^{N_\Gamma} \dot{m}_i A_{\Gamma,i}\mathcal{V}_i}{\sum_{i=1}^{N_\Gamma} A_{\Gamma,i}\mathcal{V}_i} \tag{9.44}$$

with $\alpha_{l,i}$ the volume fraction of liquid and A_Γ the interface area obtained from the PLIC reconstruction in the case of VOF or from a simplex decomposition for SLS and $\mathbf{u}_{l,i}$ components defined as average of face velocities $\mathbf{u}_l$.

9.3.1 Kinetic energy analysis

In a first attempt, the simulation is performed with the choices presented in Section 9.1.1 for VOF and SLS with special attention given on the conservation properties of mass and kinetic energy.
In Fig. 9.14d, the normalized mass $M_l/M_{l,0}$, normalized kinetic energy divided by the mass $K_l M_{l,0}/K_{l,0}M_l$ and $\dot{m}$ are represented with respect to time until the droplet is about to exit the domain. As demonstrated in Section 9.2.2, the SLS observes mass loss which remains important for the high resolution mesh with about 5% of mass loss at the end of the simulation in Fig. 9.14b. This mass loss is not due to phase change but transport errors. This has also been observed for the same test case in [29] where an ACLS method is employed to improve this behaviour. Hence, it makes difficult to quantify phase change in a dynamic simulation using SLS if the mesh is not resolved enough. For the VOF method, in Fig. 9.14a, the mass loss is driven by evaporation and seems to converge with the mesh. As for the static evaporation test case of Section 9.2.2, the evaporation is under-predicted for the coarse meshes. At this point of the analysis, the results agree with the test cases of Section 9.2.
On the other hand, the kinetic energy contained in the droplet is increasing for all the meshes excepted the high resolution mesh of VOF where it is almost constant. In fact, the kinetic energy is expected to decrease because of the drag force applied by the ambient air. The first rise of kinetic energy observed in Fig. 9.14c could be explained by the rise of $\dot{m}$ showed in Fig. 9.14d due to the initialization which is not exactly corresponding to a cylindrical Stefan flow. However, this initial rise decreases quickly and cannot be the cause of the constant rise in kinetic energy in the droplet.
Another source of kinetic energy could come from the interaction of the boundary conditions with the induced Stefan flow which can propel the droplet in the opposite side of the upper wall as demonstrated in [218]. However, this source of kinetic energy is negligible compared to the initial kinetic energy corresponding to a droplet translating at 1 m/s. It is also expected to be less present when the droplet move away from the wall. Thus it cannot



(a) *VOF* $M_l/M_{l,0}$ **(b)** *SLS* $M_l/M_{l,0}$

(c) $K_l M_{l,0}/K_{l,0} M_l$ **(d)** $\dot{m}$

Figure 9.14: *M/M_0, $K_l M_{l,0}/K_{l,0} M_l$ and $\dot{m}$ temporal evolution for VOF and SLS for the convected droplet evaporation using the standard treatment of velocity in the imposition of $\nabla \cdot \mathbf{u}^{n+1}$. The three meshes $N_D = 10$ (——), $N_D = 20$ (- - -) and $N_D = 40$ (-···-) are represented.*

explain this constant gain in kinetic energy through the simulation.
To make sure about this, the same test case has been performed without initial velocity in the droplet where the propulsion due to the Stefan flow was observed. However, the physical time required for enough transfer of momentum in the droplet to induce a significant displacement is reached far after the end of the present simulation.
This kinetic energy behaviour is then due to a source of error in the numerical treatment of the momentum.

9.3.2 Velocity jump treatment

In fact, the poor behaviour observed in this first simulation could arise from the effect of the velocity jump treatment for the imposition of $\nabla \cdot \mathbf{u}^{n+1}$. In Section 7.2.1.1, a point is made about the importance of velocity average when the momentum is solved with a WDF. In several works [195; 74; 122; 179], emphasis is put on the proper derivation of the

one-fluid equations of Chapter 2 where the velocity is mass-averaged $\mathbf{u}^M$ in the momentum equation while the divergence condition is written as a volume average $\nabla \cdot \mathbf{u}^V$. Ensuring $\nabla \cdot \mathbf{u}^{V,n+1}$ in the Poisson equation where $\nabla \cdot \mathbf{u}^{M,*}$ is obtained from mass-averaged velocities is not completely correct and can lead to inconsistent results.
In the unified framework, the momentum is treated with a WDF for both VOF and SLS leading to $\mathbf{u}^M$ while the treatment of $\nabla \cdot \mathbf{u}^{n+1}$ is done without any consideration on the average nature of the quantities. It is then interesting to implement a mass-averaged reformulation of $\nabla \cdot \mathbf{u}^{n+1}$ consistent with the prediction step to see the impact on the simulation results.
Two discretizations are proposed to solve this problem, the first comes from [179] presented in Section 7.2.1.1. The expression relies on the wetted face areas A_f and takes the form

$$\dot{s}_\rho = -\dot{m} \left[\frac{1}{\rho}\right]_\Gamma \rho_l \rho_g \left[\frac{(A_u^- - A_u^+)\, n_x}{\rho_u^- \rho_u^+} + \frac{(A_v^- - A_v^+)\, n_y}{\rho_v^- \rho_v^+}\right] \quad , \tag{9.45}$$

with superscript $-$ and $+$ corresponding to the left and right faces respectively and ρ_f^- and ρ_f^+ the wetted densities obtained from an average with the wetted areas. Then the wetted areas need to be retrieved from the simplex decomposition in the case of SLS while it is available in VOF from the PLIC reconstruction. In order to avoid such reconstruction and to be more consistent with the projection scheme, the correction Eq. (9.45) is reformulated in term of staggered density ρ_f^{n+1} and volume fractions θ_f^{n+1} summed up in Table 9.2 instead of the wetted densities and the wetted areas

$$\dot{s}_\rho = -\dot{m} \left[\frac{1}{\rho}\right]_\Gamma \rho_l \rho_g \left[\frac{(\theta_u^- - \theta_u^+)\, n_x}{\rho_u^- \rho_u^+} + \frac{(\theta_v^- - \theta_v^+)\, n_y}{\rho_v^- \rho_v^+}\right] \quad . \tag{9.46}$$

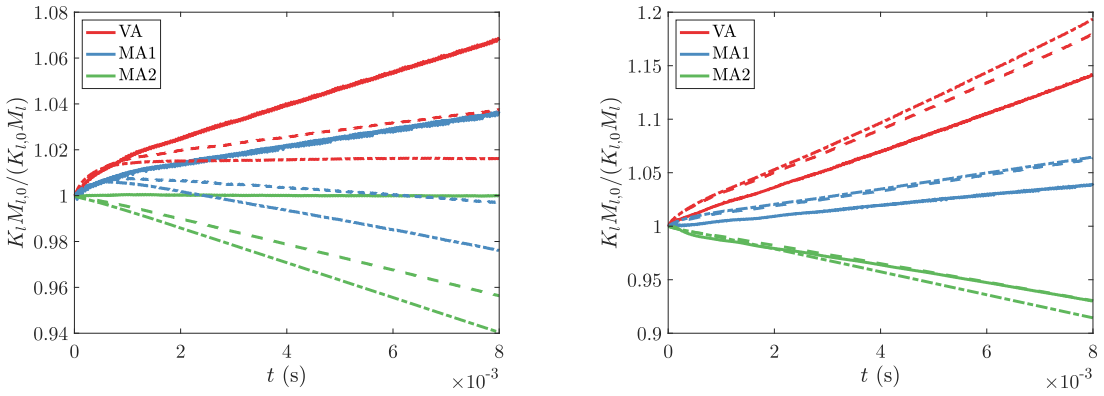
The use of ρ_f^{n+1} densities is consistent with the choice in the Poisson equation Eq. (9.1) where it appears as the variable coefficient.
Then all the formulations are compared on the convected droplet test case in Fig. 9.15:

- The initial velocity jump treatment which is volume averaged (VA)
- The expression of Eq. (9.45) used in [179] (MA1)
- The expression of Eq. (9.46) proposed in this work (MA2)

It turns out that the reformulations MA1 and MA2 have an important impact on the momentum transfers at the interface. While the MA1 seems to already fix the problem in VOF in Fig. 9.15a, it reduces the spurious kinetic energy increase in the case of SLS in Fig. 9.15b even if the tendency is still not the one expected. On the other hand, the new formulation MA2 consistent with the Poisson equation allows the SLS method to lose kinetic energy instead of artificially gaining. This last formulation is then the one used for a deeper study of this test case.

9.3.3 Final methodology

First, snapshots for different times are presented in Fig. 9.16 for the medium mesh. The temperature remains very high at the head of the droplet leading to a strong vaporization rate while at the trail, a smoother evolution is observed leading to a vaporization rate



(a) *VOF* $K_l M_{l,0}/K_{l,0} M_l$ **(b)** *SLS* $K_l M_{l,0}/K_{l,0} M_l$

Figure 9.15: *$K_l M_{l,0}/K_{l,0} M_l$ temporal evolution for VOF (left) and SLS (right) for the convected droplet evaporation using three different methodology in the imposition of $\nabla \cdot \mathbf{u}^{n+1}$. The three meshes $N_D = 10$ (——), $N_D = 20$ (- - -) and $N_D = 40$ (-···-) are represented.*

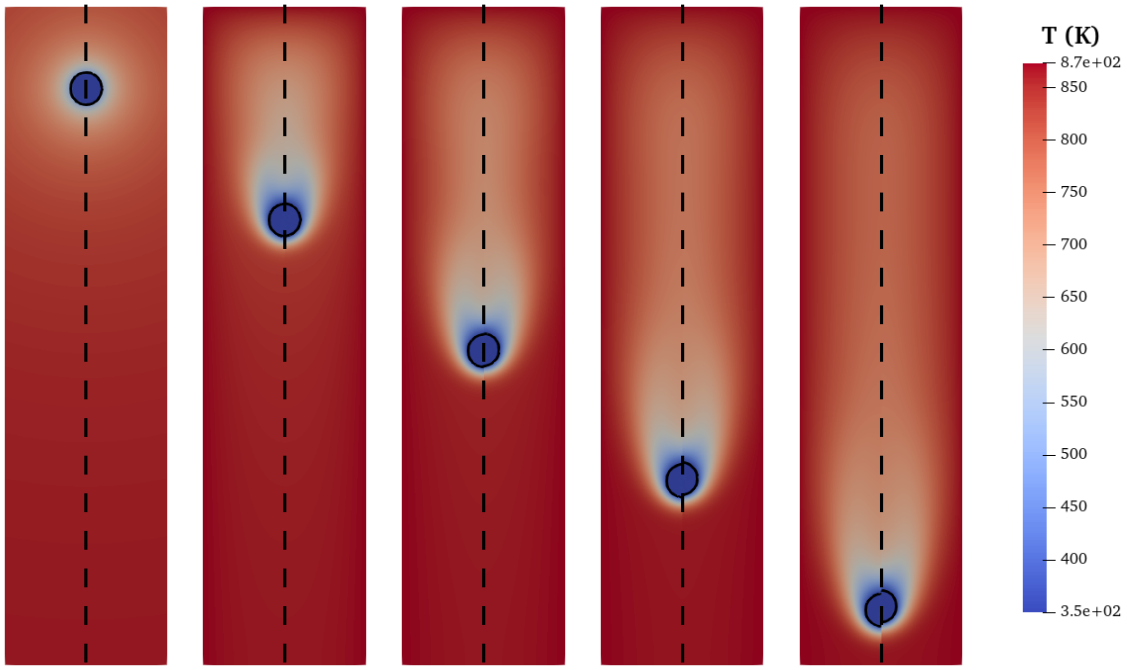
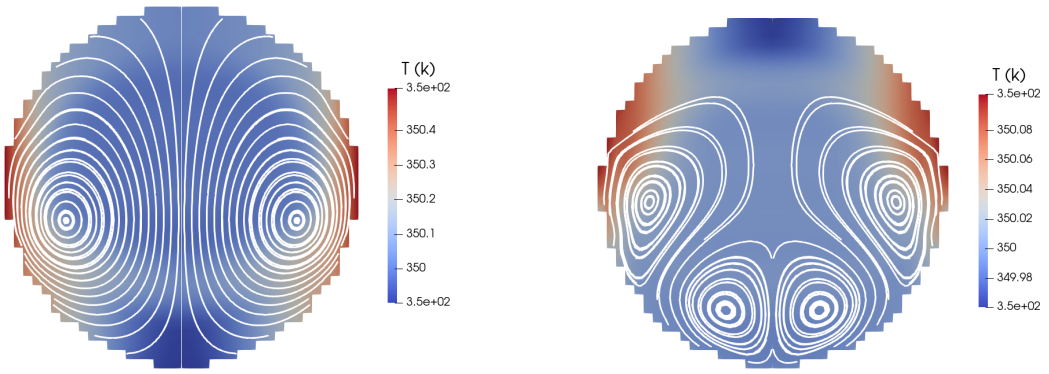


Figure 9.16: *Snapshots of the convected droplets at different times (interval of 0.002 s) with the medium mesh. The VOF method is at the left while SLS is at the right.*

closer to a static evaporation problem. Note that the main difference observed here is the velocity of the droplet which is faster for VOF than for SLS. This can be explained by the difference of conservation properties of the capturing methods demonstrated in Chapter 4 and Chapter 6.



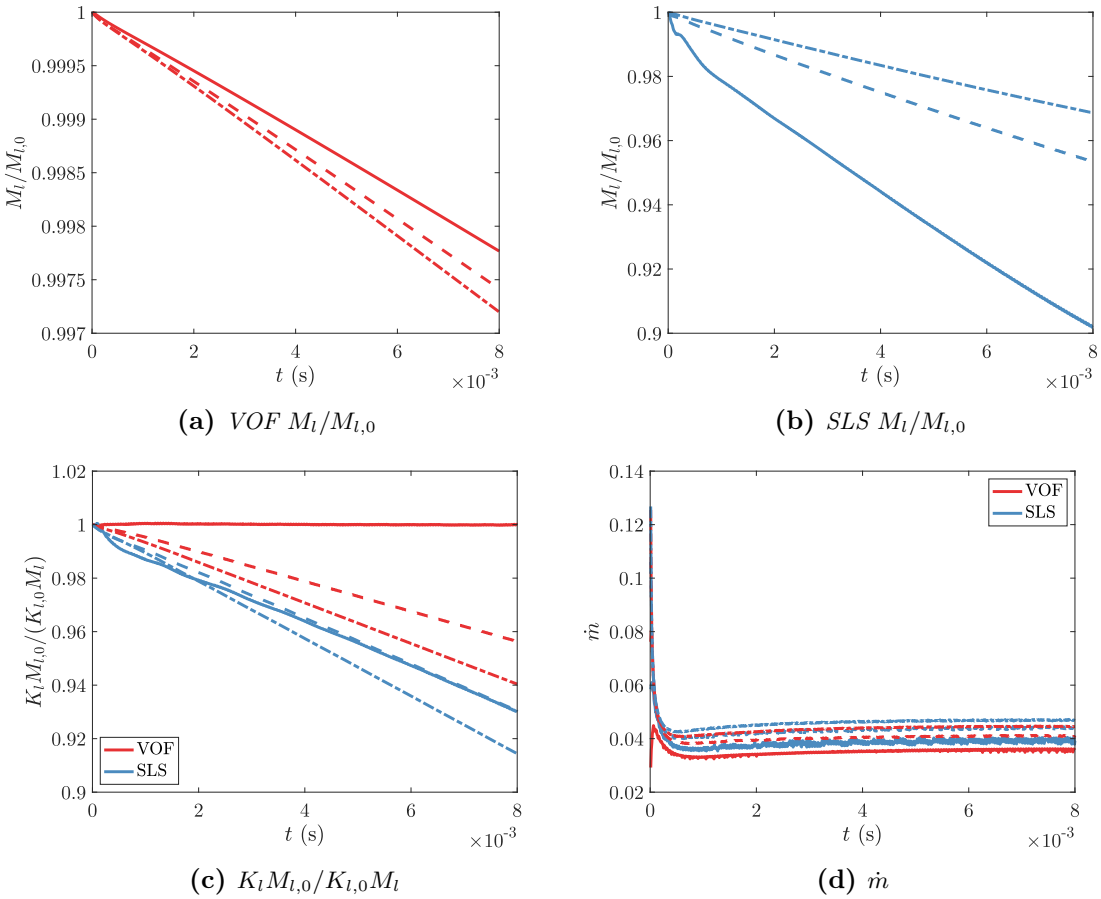
(a) *VOF* **(b)** *SLS*

Figure 9.17: *Zoom on the droplet for the highest mesh* $N_D = 40$.

Then, a zoom is performed in the droplet to observe the recirculation and the liquid temperature in Fig. 9.17. As in [183], a recirculation is observed for the VOF droplet in Fig. 9.17a which is expected to produce a lower temperature at the center of the vortex. This is not clearly observed here but the temperature difference is under 0.1% and variations might not be well-captured with this mesh size. However, the SLS shows two vortices in Fig. 9.17b which might arise from spurious transfers of kinetic energy at the interface.
For a closer look at the mass and momentum conservation, the same temporal evolutions as for the first attempt are plotted in Fig. 9.18
The modification in the velocity jump treatment does not seem to impact drastically the other metrics. Indeed, the mass loss is still driven by the transport errors in the case of SLS in Fig. 9.18b while it shows a convergence behaviour for VOF in Fig. 9.18a. In Fig. 9.18d, the evaporation rate shows similar values to the one obtained with the VA method. Hence, the issue of using a proper velocity jump formulation would not have been detected without looking at the kinetic energy metric in this specific test case.
Finally, some metrics are provided about the evaporation rate spatial distribution. As the convection effect blow the vapour from the head of the droplet, the temperature and species field are non-uniform leading to a heterogeneous evaporation rate along the droplet azimuthal angle. The azimuthal profile of $\dot{m}$ is plotted in Fig. 9.19 for the medium mesh at different times corresponding to the snapshot times of Fig. 9.16. It is clear that the phase-change procedure is able to represent non-uniform evaporation starting from a static Stefan flow problem. Both VOF and SLS shows similar tendency with a slightly superior evaporation rate for SLS. Moreover SLS is able to provide a perfectly uniform initial azimuthal profile of $\dot{m}$ while VOF shows a little slope because of gradient reconstruction errors.

9.3.4 Conclusion

This last test case enlightens the importance of using numerical schemes consistent with the initial two-phase flow derivation used. The ambiguity is more critical when using a WDF as the quantities have a different meaning depending on the equation considered. As

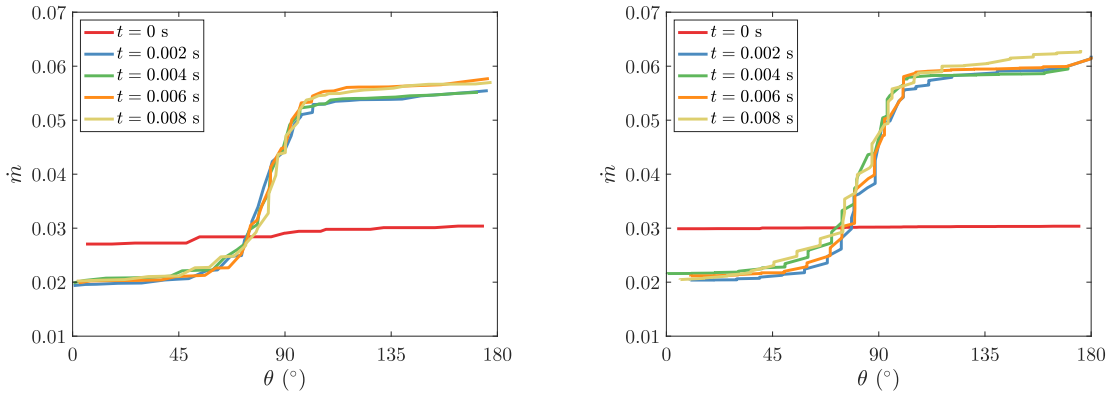


(a) *VOF* $M_l/M_{l,0}$ **(b)** *SLS* $M_l/M_{l,0}$

(c) $K_l M_{l,0}/K_{l,0} M_l$ **(d)** $\dot{m}$

Figure 9.18: *M/M_0, $K_l M_{l,0}/K_{l,0} M_l$ and $\dot{m}$ temporal evolution for VOF and SLS for the convected droplet evaporation using MA2. The three meshes $N_D = 10$ (——), $N_D = 20$ (- - -) and $N_D = 40$ (-···-) are represented.*

the projection method implies a tight coupling between the continuity equation and the momentum equation through the Poisson equation, it is then crucial to make sure that all operators and quantities are written using the same averages. The correct derivation of $\nabla \cdot \mathbf{u}^{n+1}$ has shown to have a huge impact on the momentum transfer at the interface.
Finally, the use of WDF applied to SLS seems to be inappropriate for accurate simulations of two-phase flow with phase change as it cannot correctly reproduce the physics of the flow in the droplet. This was not observed in the study of Chapter 6 as the WDF and JCF are almost equivalent when velocity is continuous across the interface. In the case of phase change, the velocity jump implies additional averaging of quantities when using a WDF, which are not straightforward to handle using SLS. One explanation arises from the derivation itself in Chapter 2 where it has been shown that the transport of ϕ is not related to the transport of the indicator function. Then, the terms at the interface are closed through ad-hoc considerations. Thus, the JCF seems to be the most-suited choice for SLS



(a) *VOF* **(b)** *SLS*

Figure 9.19: *Azimuthal profile of $\dot{m}$ for different time.* 0° *corresponds to the trail of the droplet while* 180° *is the head of the droplet.*

as it does not require closures but only the accurate imposition of jump and boundary conditions at the interface. This is in agreement with the literature where WDF and JCF applied to SLS have been compared in [219] for boiling where JCF showed superior results.

9.4 Conclusion

To conclude this last chapter, the extension of the uniform framework has been presented, evaluated on canonical test cases and demonstrated on a convected droplet. The different elements of the phase-change solver have been chosen with special attention given on the reconstruction of $\dot{m}$ and coupling of the governing equations. The numerical results presented here lead to interesting key points on the phase-change procedure.

First, the use of quadratic extrapolation seems to substantially improve the accuracy of the solver even if the scalar fields are at most second-order accurate. This allows to making systematic choice for phase-change simulations using the unified framework. Nevertheless, in multidimensional configurations, the errors arising from topological representations of the interface can become predominant in the computation of $\dot{m}$, and in such a case, the use of linear extrapolation seems to provide more accurate results for VOF. It can be explained by analogy to the curvature considerations made in Chapter 6. As SLS provides a more accurate topology representation compared to VOF, it is less prompt to errors in the reconstruction of $\dot{m}$.

The second point raised in the numerical results is the importance to reduce as much as possible the coupling between the different scalar equations to obtain a robust and consistent phase-change procedure. More specifically, it has been demonstrated that some choice combinations do not lead to convergent simulations of phase change even in 1D.

Finally, the mass loss through reinitialization is prohibitive for correct evaporation estimations. Luckily, this mass loss is not predominant when a sufficient resolution is used. But this implies a serious constraint on the mesh size if several droplets are to be investigated

in a finite domain.
From all these considerations, an application to a convected droplet has been presented with mesh convergence where the mass loss is more severe than for a static case. Again, this shows the limits of SLS for evaporation simulations when the resolution is not important enough. In this test case, the momentum conservation has also shown discrepancies in the results of VOF and SLS. While VOF conserves momentum, SLS cannot maintain the droplet velocity and a shift between VOF and SLS droplet position increases with time.
This last test case also brought out the issues related to a WDF applied to SLS. Even using a correction on the velocity jump treatment that is consistent with the WDF of velocity, the results are still not in agreement with the expected physical behaviour of the velocity field. It is then clear that JCF is the most-suited method for two-phase simulations with phase change in light of the numerical experiments presented in this chapter and the literature.
To circumvent the issue of accuracy loss in the evaluation of $\dot{m}$ for VOF and the mass and momentum conservation issues for the SLS, a next step could be to adapt the CLSVOF method to the extended unified framework.

Conclusion and perspectives

Conclusions on the present work

The work presented in this manuscript proposes a step-by-step methodology to build an incompressible two-phase flow solver with phase change based on a sharp interface method. The solver is built sequentially by first transporting the interface, then coupling it to an incompressible solver without phase change and finally adding the phase change in the solver.

First, the general governing equations of fluid mechanics were applied to evaporating incompressible two-phase flows using an infinitely thin and massless interface representation in Chapter 1.
Then, Chapter 2 showed the derivation of the two-phase flow equations applied to a finite-volume framework with emphasis on the closure of the different contribution terms at the interface. The VOF method naturally arises from the finite-volume average of the two-fluid equations, while the LS method results in numerical closures introduced to avoid issues with the transport of the volume fraction. The LS method then constitutes a framework with much degree of freedom on the numerical closures at the interface with the disadvantage of losing conservation properties on the mass, momentum and energy. While this observation directly comes from the proper derivation of the equations, it has been widely investigated through numerical experiments with increasing complexity at different stages of the manuscript. At this point in the manuscript, the choice of an interface capturing method to solve the set of equations was unclear and needed to be investigated.

For this aim, Chapter 3, a thoroughgoing review of interface capturing methods was provided, resulting in the choice of four interface capturing methods that are worth comparing: VOF, SLS, ACLS and CLSVOF. They all expose advantages and drawbacks regarding conservation properties, topology accuracy, efficiency and implementation complexity.
Chapter 4 presented a comparison of these four interface capturing methods on imposed velocity fields. The VOF and CLSVOF methods showed the best results in shape accuracy and mass conservation as they are inherently conservative, while SLS provided comparable results with ACLS. Moreover, the use of SLS implies a significant decrease in computational time and implementation complexity compared to ACLS. At this stage, VOF and CLSVOF seemed to be the most promising methods for two-phase flow simulations. However, they needed to be evaluated in more relevant flow configurations where the interface is coupled to the flow.

To this aim, the methodology to couple the interface to an incompressible solver without phase change was detailed in Chapter 5. It focused on the conservation of momentum and the modelling of surface tension. At the end of this chapter, a unified framework was presented to adapt the four interface capturing methods to a robust and consistent two-phase flow solver. This framework includes a sharp transport of the interface, a well-balanced surface tension discretization and a consistent mass and momentum transport, which allows capillary-driven simulations with high-density ratios.
This solver was evaluated using the four interface capturing methods in Chapter 6. The analysis was separated into multiple test cases devoted to the different points of attention disclosed in Chapter 5. First, the magnitude of spurious currents for both static and dynamic test cases was quantified. It led to interesting results regarding the accuracy of the curvature computation for the different methods. SLS provided the most accurate curvature, and is able to keep very low spurious currents even in a dynamic configuration, while all other methods provided non-convergent behaviours in this challenging test case. Then the momentum conservation was evaluated with the density ball translation. VOF and CLSVOF showed momentum conservation of about one order of magnitude better than SLS. On the other hand, ACLS improved conservation compared to SLS but still remained less conservative than VOF and CLSVOF. Finally, a droplet collision and a shear layer were presented to demonstrate the feasibility of the solver on complex configurations. In these applications, the most promising capturing method was the CLSVOF, as it was able to conserve mass and provide robust computation of the curvature.

Finally, the last part of this manuscript was dedicated to the incorporation of phase change in the solver. In Chapter 7, the different challenges related to phase-change simulations were presented based on a state-of-the-art. In fact, phase change implies solving the energy equation and the species mass fraction of vapour in the gas with associated flux jumps. Moreover, the velocity is discontinuous at the interface and requires special numerical treatments. Finally, the interface capturing method needs to take correctly into account the regression due to phase change in its numerical methods. All these challenges led to investigate the reconstruction of the evaporation rate and other quantities at the interface. This was the topic of Chapter 8 where scrupulous numerical experiments were performed to analyze and choose well-suited reconstruction methods for the quantities at the interface. Using an SLS method, the most accurate way to reconstruct gradients at the interface relies on a ghost cell method, while VOF methods provide accurate gradients by using functional fitting through least-square minimization. Both methods can either base their reconstruction on linear or quadratic extrapolated fields. The numerical experiments did not allow to conclude on the most-suited type of extrapolation when applied to fields that are already entailed with numerical errors.
The study of planar and spherical Stefan flow problems in Chapter 9 allowed to show that in most of the cases, the quadratic extrapolation provided the most accurate results. In this same chapter, a numerical study showed that the coupling between energy and species equations had to be done with care to obtain a consistent phase-change procedure. From all these considerations, a solver based on VOF and SLS has been built and tested on the

convection of an evaporating droplet. This last test case exposed momentum conservation issues when the velocity jump was ensured numerically. The source of this problem was detected and arisen from the two-phase derivation used in this work based on a whole domain formulation. It also showed that a whole domain formulation applied on SLS did not constitute an adapted methodology for accurate phase-change simulations.

Overall, the work of this manuscript leads to a better understanding of numerical methods applied to the two-phase flow governing equations. The numerical experiments have demonstrated the complexity and subtlety in the manipulation of numerical methods to solve two-phase flows with phase change. The constraint imposed by this framework exposes the intimate relation between the choice of an interface representation (the interface capturing method) and the equation formulation (the closure of the contributions at the interface).
As VOF naturally derives from the two-fluid and one-fluid formulations applied to a finite-volume framework, the closure of the interface contribution in the governing equations can be written directly. The derivation of the same one-fluid and two-fluid equations for LS is way more challenging and has not been presented yet in the literature. Instead, LS makes use of assumptions to close the system of equations that are based on numerical considerations. This gives the illusion that the closure of interface contributions comprises a large degree of freedom when using LS. This idea is even confirmed by the test cases presented without phase change where all interface capturing methods were performing well and provided the same physical results.
However, this manuscript have exposed the numerical issues arising from the use of the unified framework with LS when phase change occurs. Indeed, when the same framework is applied to VOF and LS, the results highly depends on the handling of the velocity jump in the continuity equation and it is then difficult to close this term when LS is using a whole domain formulation. It comes to the conclusion that the addition of phase change introduces implicitly more numerical constraints than it seems. VOF can use already well-known derivations of one-fluid and two-fluid formulations, while LS relies better on a jump condition formulation with appropriate numerical closures.

Perspectives

The first perspectives of this manuscript revolve around the improvement of the numerical methods applied to two-phase simulations with phase change. This can come from two different lines of attack:

- A better understanding of the mathematical derivation of the governing equations for two-phase flows using the Level-Set class of methods.
- An improvement of interface transport and reconstruction of interface quantities from numerical considerations.

Indeed, the conclusions of the manuscript encourage to clarifying the derivations of the equations using LS. With a proper derivation of two-fluid and one-fluid formulations adapted to the transport of LS, the ambiguity on interface contribution closures is removed. Then,

the numerical issues arising from these inconsistencies could be avoided, which significantly improves the robustness and accuracy of the numerical methods associated.
Then, the solver presented in this manuscript could take into account further adjustments of the literature to improve the behaviour of interface capturing methods in the mesh limit. For example, the works on the redistanciation presented in Chapter 3 could be integrated into the solver to limit the mass loss induced by numerical errors when using SLS. Moreover, SLS could make use of a jump condition formulation to alleviate the closure issues related to the mass transfer at the interface as in [183]. On the other hand, CLSVOF has been promising in the study of two-phase flows without phase change and its extension to phase change could lead to a powerful methodology able to provide the best of the two worlds: VOF conservation properties and SLS accurate reconstructions. This idea is not new and has already been done for boiling applications in [220; 147] and for evaporation processes in [126] with encouraging results.

From the perspective of studying droplet evaporation in turbulent flows, the present methodology finds some limits because of the incompressibility assumption. It is seriously limiting as it requires to have open boundaries for the well-posedness of the physical problem. This can be treated by injecting turbulence from an inlet as performed in the PhD thesis of Alis [4] or modifying the global gas density as proposed in [159]. However, to widen the range of possibilities for the physical configurations, the low Mach formulation of the equations could be used instead of the incompressible system considered here. The inclusion of compressibility in the low Mach solvers showed a recent growth of interest with different works such as the simulation of evaporation droplets in HIT in [52] where the gas is supposed to be compressible. Other works also take into account the compressibility of the liquid, which allows for bubble inclusions in [126]. Finally, a pressure-based all-Mach solver has been developed recently to perform simulations of supersonic jets [101].

Finally, another perspective arises directly from the framework of this research project which encompasses both sharp and diffused interface methods for the study of two-phase flows. The derivation of VOF two-fluid has a lot of similarities with the multifluid models, and ideas from the sharp interface methods to reconstruct accurate quantities at the interface could be used for the closure of the multifluid equations. Some developments in this vein have been presented in [45] where a low-Mach multifluid model is derived and applied to two-phase flow simulations with mass transfer.

Appendix A

Ghost Fluid Method

A.1 General ghost fluid method

The Ghost Fluid Method (GFM) was first introduced by Fedwik et al. [59] for treating multi material flows. It has been generalized later to the variable coefficient Poisson equation with jump conditions in [117]. Let us consider a quantity Φ with a variable coefficient β in a domain $\Omega = \Omega_l \cup \Omega_g$ with jump at the interface Γ in the zeroth and first order normal derivatives a_Γ and b_Γ respectively. β can be discontinuous at the interface. However it is supposed continuous in Ω_l and Ω_g. For the present work, it is supposed piecewise constant with a value in the liquid β_l and a value in the gas β_g. Then, the Poisson equation applied to Φ reads

$$\nabla \cdot (\beta \nabla \Phi) = S \qquad \text{in } \Omega \quad , \tag{A.1}$$

$$[\Phi]_\Gamma = a_\Gamma \qquad \text{at } \Gamma \quad , \tag{A.2}$$

$$[\beta \nabla \Phi \cdot \mathbf{n}_\Gamma]_\Gamma = b_\Gamma \qquad \text{at } \Gamma \quad , \tag{A.3}$$

with $\mathbf{n}_\Gamma$ the normal pointing outside the interface (from the liquid to the gas phase) and S a source term.
If $a_\Gamma = b_\Gamma = 0$, then a classical second order discretization of Eq. (A.2) for a given computational cell $\mathcal{C}_i$ leads to

$$\frac{\beta_{i+\frac{1}{2}} \left.\frac{\partial \Phi}{\partial x}\right|_{i+\frac{1}{2}} - \beta_{i-\frac{1}{2}} \left.\frac{\partial \Phi}{\partial x}\right|_{i-\frac{1}{2}}}{\Delta x} = S_i \quad , \tag{A.4}$$

with $\left.\frac{\partial \Phi}{\partial x}\right|_{i+\frac{1}{2}} = \frac{\Phi_{i+1} - \Phi_i}{\Delta x}$, $\left.\frac{\partial \Phi}{\partial x}\right|_{i-\frac{1}{2}} = \frac{\Phi_i - \Phi_{i-1}}{\Delta x}$.
In the presence of jump at the interface, GFM introduces the notion of ghost values Φ^{gh} such that at each point in the domain

$$\Phi^{gh}_{i,l} = \Phi_{i,g} + a_i \quad \text{if } \mathcal{C}_i \in \Omega_g \quad . \tag{A.5}$$

$$\Phi^{gh}_{i,g} = \Phi_{i,l} - a_i \quad \text{if } \mathcal{C}_i \in \Omega_l \quad . \tag{A.6}$$

with a_i the interface jump defined at the cell center $\mathbf{x}_i$. A Taylor series expansion of $[\Phi]_\Gamma$ around $\mathbf{x}_\Gamma$ at an arbitrary point $\mathbf{x}$ leads to

$$[\Phi](\mathbf{x}) = [\Phi]_\Gamma + (\mathbf{x}-\mathbf{x}_\Gamma)\cdot[\nabla\Phi]_\Gamma + \mathcal{O}\left((\mathbf{x}-\mathbf{x}_\Gamma)^2\right) \quad . \tag{A.7}$$

Then, rewriting the first derivative jump

$$[\beta\nabla\Phi]_\Gamma = \beta_l\,\nabla\Phi_l|_\Gamma - \beta_g\,\nabla\Phi_g|_\Gamma = \beta_l\,[\nabla\Phi]_\Gamma + [\beta]_\Gamma\,\nabla\Phi_g|_\Gamma \quad . \tag{A.8}$$

The same decomposition can be written in term of β_g and $\nabla\Phi_l$ which leads to 2 different definitions for $[\nabla\Phi]_\Gamma$:

$$[\nabla\Phi]_\Gamma = \frac{1}{\beta_l}[\beta\nabla\Phi]_\Gamma - \frac{[\beta]_\Gamma}{\beta_l}\nabla\Phi_g|_\Gamma \quad , \tag{A.9}$$

$$= \frac{1}{\beta_g}[\beta\nabla\Phi]_\Gamma - \frac{[\beta]_\Gamma}{\beta_g}\nabla\Phi_l|_\Gamma \quad . \tag{A.10}$$

Let us consider now that $\mathcal{C}_i \in \Omega_l$ and $\mathcal{C}_{i+1} \in \Omega_g$. In this case, the face flux from the gas perspective is written as

$$\beta_{i+\frac{1}{2}}\left.\frac{\partial\Phi}{\partial x}\right|_{i+\frac{1}{2}} = \beta_g\left.\frac{\partial\Phi}{\partial x}\right|_{g,i+\frac{1}{2}} = \beta_g\frac{\Phi_{g,i+1}-\Phi^{gh}_{g,i}}{\Delta x} = \beta_g\frac{\Phi_{g,i+1}-\Phi_{l,i}+a_i}{\Delta x} \quad , \tag{A.11}$$

and Eqs. (A.7) and (A.9) in that configuration yields:

$$a_i = a_\Gamma - \frac{\theta\Delta x}{\beta_l}b_\Gamma + \frac{\theta\Delta x\,[\beta]_\Gamma}{\beta_l}\left.\frac{\partial\Phi}{\partial x}\right|_{g,i+\frac{1}{2}} \quad , \tag{A.12}$$

with $\theta = \frac{x_\Gamma - x_i}{\Delta x}$.
By using this last expression in Eq. (A.11), one can write

$$\beta_g\left.\frac{\partial\Phi}{\partial x}\right|_{g,i+\frac{1}{2}} = \beta_g\frac{\Phi_{g,i+1}-\Phi_{l,i}}{\Delta x} + \beta_g\frac{a_\Gamma}{\Delta x} - \frac{\theta\beta_g}{\beta_l}b_\Gamma + \frac{\theta\,[\beta]_\Gamma}{\beta_l}\beta_g\left.\frac{\partial\Phi}{\partial x}\right|_{g,i+\frac{1}{2}} \quad . \tag{A.13}$$

This last equation can be rearranged to obtain an expression for the face flux

$$\beta_g\left.\frac{\partial\Phi}{\partial x}\right|_{g,i+\frac{1}{2}} = \widehat{\beta}\frac{\Phi_{g,i+1}-\Phi_{l,i}}{\Delta x} + \widehat{\beta}\frac{a_\Gamma}{\Delta x} - \frac{\theta\widehat{\beta}}{\beta_l}b_\Gamma \quad , \tag{A.14}$$

with $\widehat{\beta}$ an effective β value defined as

$$\widehat{\beta} = \frac{\beta_g\beta_l}{\beta_g\theta + \beta_l(1-\theta)} \tag{A.15}$$

The same relation can be written from the liquid perspective

$$\beta_l\left.\frac{\partial\Phi}{\partial x}\right|_{l,i+\frac{1}{2}} = \widehat{\beta}\frac{\Phi_{g,i+1}-\Phi_{l,i}}{\Delta x} + \widehat{\beta}\frac{a_\Gamma}{\Delta x} + \frac{(1-\theta)\widehat{\beta}}{\beta_g}b_\Gamma \tag{A.16}$$

It is interesting to show that differencing Eqs. (A.14) and (A.16) yields exactly the correct jump in the first derivative b_Γ

$$\begin{aligned}\beta_l \left.\frac{\partial \Phi}{\partial x}\right|_{l,i+\frac{1}{2}} - \beta_g \left.\frac{\partial \Phi}{\partial x}\right|_{g,i+\frac{1}{2}} &= \frac{(1-\theta)\widehat{\beta}}{\beta_g} b_\Gamma - \frac{\theta\widehat{\beta}}{\beta_l} b_\Gamma \\ &= \left(\frac{(1-\theta)\widehat{\beta}}{\beta_g} - \frac{\theta\widehat{\beta}}{\beta_l}\right) b_\Gamma = b_\Gamma \quad . \end{aligned} \tag{A.17}$$

A.2 Discrete equivalence between WDF and JCF for surface tension

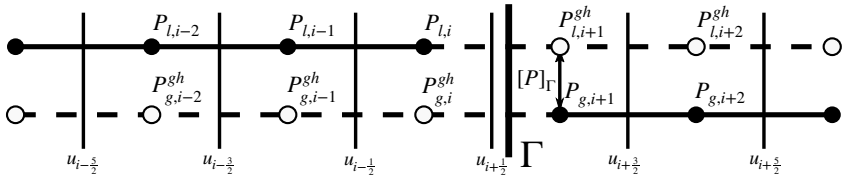


Figure A.1: *Ghost fluid representation of pressure*

Let us show that WDF and the JCF are equivalent for surface tension modelling if no phase-change occurs.
Rewriting Eq. (5.3) at x_i for the 1D case represented in Fig. A.1 gives

$$\frac{\frac{1}{\rho_{i+\frac{1}{2}}} \frac{\partial P}{\partial x}\big|_{i+\frac{1}{2}} - \frac{1}{\rho_{i-\frac{1}{2}}} \frac{\partial P}{\partial x}\big|_{i-\frac{1}{2}}}{\Delta x} = \frac{1}{\Delta t} \left.\frac{\partial u^*}{\partial x}\right|_i \quad . \tag{A.18}$$

The interface lies between P_i and P_{i+1} with liquid to the left, hence, one can inject Eq. (5.50) into Eq. (A.18) leading to

$$\frac{\frac{1}{\rho_{i+\frac{1}{2}}} \frac{P_{i+1}-P_i}{\Delta x} - \frac{1}{\rho_{i-\frac{1}{2}}} \frac{P_i - P_{i-1}}{\Delta x}}{\Delta x} + \frac{1}{\rho_{i+\frac{1}{2}}} \frac{[P]_\Gamma}{\Delta x^2} = \frac{1}{\Delta t} \frac{u^*_{i+\frac{1}{2}} - u^*_{i-\frac{1}{2}}}{\Delta x} \quad . \tag{A.19}$$

By rewriting the jump term and noticing that the first term corresponds to the Laplacian operator in a WDF

$$\nabla \cdot \left(\frac{1}{\rho} \nabla P\right)\Big|^{WDF} = \frac{\frac{1}{\rho_{i+\frac{1}{2}}} \frac{P_{i+1}-P_i}{\Delta x} - \frac{1}{\rho_{i-\frac{1}{2}}} \frac{P_i - P_{i-1}}{\Delta x}}{\Delta x} \quad , \tag{A.20}$$

it follows

$$\nabla \cdot \left(\frac{1}{\rho} \nabla P\right)\Big|^{WDF} = \frac{1}{\Delta t} \frac{\left(u^*_{i+\frac{1}{2}} - \frac{\Delta t}{\rho_{i+\frac{1}{2}}} \frac{[P]_\Gamma}{\Delta x}\right) - \left(u^*_{i-\frac{1}{2}} - 0\right)}{\Delta x} \quad . \tag{A.21}$$

Finally, one can show that $u^*_{i+\frac{1}{2}} - \frac{\Delta t}{\rho_{i+\frac{1}{2}}} \frac{[P]_\Gamma}{\Delta x} = u^{*,WDF}_{i+\frac{1}{2}}$ with $\delta_\Gamma \mathbf{n}_\Gamma|_{i+\frac{1}{2}} = -\frac{1}{\Delta x}$ and $u^*_{i-\frac{1}{2}} - 0 = u^{*,WDF}_{i-\frac{1}{2}}$ with $\delta_\Gamma \mathbf{n}_\Gamma|_{i-\frac{1}{2}} = 0$.
It is the proof that

$$\nabla \cdot \left(\frac{1}{\rho}\nabla P\right)\Big|^{JCF} - \frac{1}{\Delta t}\nabla \cdot \mathbf{u}^{*,JCF} = \nabla \cdot \left(\frac{1}{\rho}\nabla P\right)\Big|^{WDF} - \frac{1}{\Delta t}\nabla \cdot \mathbf{u}^{*,WDF} \tag{A.22}$$

Appendix B

Normal extrapolation

Contents

B.1 Overview of the methods

When dealing with phase-change simulations, some quantities are very local to the interface. Most of the time, their accurate computation is only possible in Ω_Γ. However they can be needed in a narrow band around the interface to avoid numerical issues.
It is also useful to define ghost values in the other phase from extrapolation instead of using the original GFM approach. For example, the temperature field T cannot rely on the temperature jump at the interface to define ghost values as $[T]_\Gamma = 0$. Then, extrapolation is needed to define ghost values in the other phase.
This extrapolation is usually performed normal to the interface as it is the natural direction of propagation of an information from the interface.

B.1.1 PDE approach

Aslam [9] introduced a multi-dimensional extrapolation approach relying on the resolution of successive PDEs. The idea is to extrapolate quantities of a known region (here the liquid) Ω_l to an unknown region (here the gas region) Ω_g separated by an interface Γ.

Constant extrapolation The constant extrapolation of a quantity Φ from Ω_l to Ω_g can be achieved solving the following PDE in pseudo time τ

$$\frac{\partial \Phi}{\partial \tau} + \chi_g \mathbf{n} \cdot \nabla \Phi = 0 \tag{B.1}$$

this results in the propagation of the quantity Φ at the speed of the normal $\mathbf{n}$ only in Ω_g. The indicator function χ_g is defined as

$$\chi_g(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \Omega_l \\ 1 & \text{if } \mathbf{x} \in \Omega_g \end{cases} \tag{B.2}$$

Φ is preserved in the known region and information is only propagated in the unknown region.

Linear extrapolation In the previous paragraph, the constant extrapolation has been presented where a PDE had to be solved in pseudo-time. Now, the procedure for a linear extrapolation is presented where the normal slope of Φ is also taken into account for the propagation in the normal direction. The PDE described by Eq. (B.1) is slightly modified to force the propagation with a slope instead of a constant propagation

$$\frac{\partial \Phi}{\partial \tau} + \chi_g \left(\mathbf{n} \cdot \nabla \Phi - \Phi^{(1)} \right) = 0 \tag{B.3}$$

with $\Phi^{(1)}$ the first derivative in the normal direction $\frac{\partial \Phi}{\partial n}$. The source term $\Phi^{(1)}$ is a forcing term such that the steady state of Eq. (B.3) respects $\mathbf{n} \cdot \nabla \Phi = \Phi^{(1)}$.
As $\Phi^{(1)}$ is not initially available in Ω_g, it is also deduced from Ω_l through Eq. (B.1) but applied to $\Phi^{(1)}$

$$\frac{\partial \Phi^{(1)}}{\partial \tau} + \chi_g \mathbf{n} \nabla \Phi^{(1)} = 0 \tag{B.4}$$

this results in the propagation of the quantity $\Phi^{(1)}$ at the speed of the normal $\mathbf{n}$ only in Ω_g.

Arbitrary order extrapolation The same procedure can be applied to an arbitrary high order extrapolation of Φ. One need to solve the successive PDEs defined by

$$\frac{\partial \Phi^{(q)}}{\partial \tau} + \chi_g \left(\mathbf{n} \cdot \nabla \Phi^{(q)} - \Phi^{(q+1)} \right) = 0 \tag{B.5}$$

with q the order of the derivative. To obtain a m order extrapolation, m successive PDEs need to be solved. The first PDE is solved for $\Phi^{(m)}$ imposing $\Phi^{(m+1)} = 0$ leading to a constant extrapolation of the m^{th} order derivative of Φ.

Numerical discretization This method requires to define a normal $\mathbf{n}$ in the whole domain or at least in a narrow band where the quantity is needed in the unknown domain. In a Level Set framework, the normal is naturally defined everywhere through differentiation of ϕ from Eq. (5.56). However, for a VOF framework, the normal $\mathbf{n}_\Gamma$ is only defined in Ω_Γ and has to be retrieved in pure cells to apply such PDE approach. This can be done by constructing a signed distance function from the isocontour $f^{-1}(0.5)$ as in [192; 27]. Another approach presented in the PhD Thesis of Kunkelmann [102] passes the data from the mixed cells to the neighbouring cells by simple average iteratively. The algorithm is

presented hereafter in Appendix B.1.3.
Then, the normal derivatives are retrieved successively using centred finite differences

$$\Phi^{(q+1)} = \mathbf{n} \cdot \nabla \Phi^{(q)} \tag{B.6}$$

In this expression, only known values of $\Phi^{(q)}$ are used. It means that the numerical representation of χ_g varies with q. To illustrate this, a quadratic extrapolation of Φ in Ω_g is represented in Fig. B.1. The function Φ is available at the left of Γ from $\mathcal{C}_{i-3}$ to $\mathcal{C}_i$.

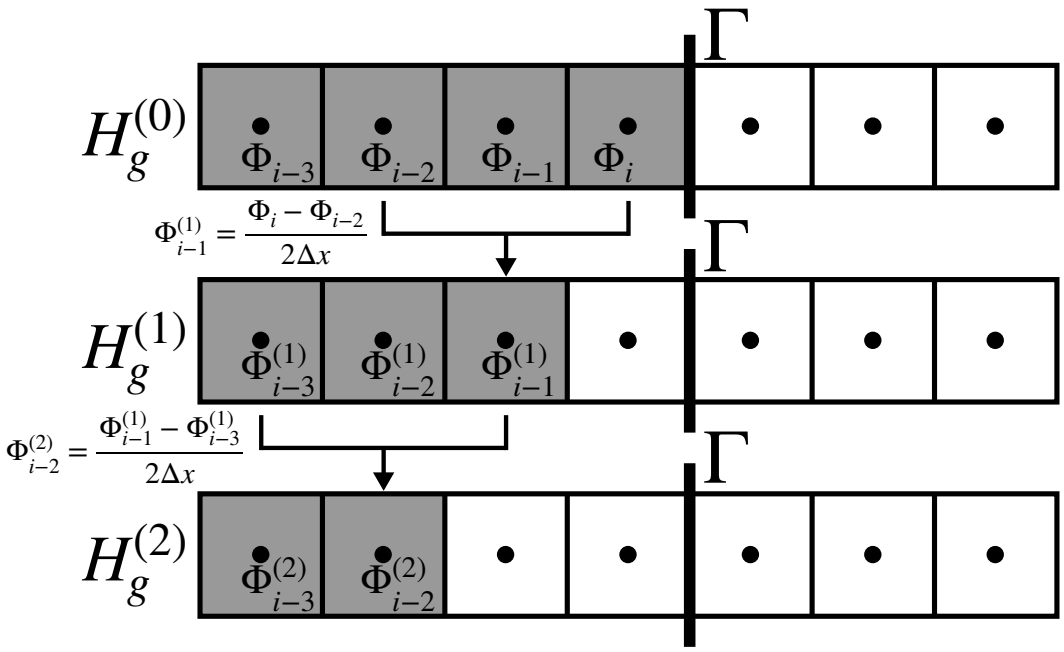


Figure B.1: *Definition of the successive $H_g^{(q)}$ for a quadratic extrapolation*

The computation of $\Phi^{(1)}$ requires only available values of Φ in the centred difference stencil. $\Phi^{(1)}$ is then only defined in the cells from $\mathcal{C}_{i-3}$ to $\mathcal{C}_{i-1}$.
For the second order derivative $\Phi^{(2)}$ requires only available values of $\Phi^{(1)}$ and can be computed from $\mathcal{C}_{i-3}$ to $\mathcal{C}_{i-2}$.
Overall, this means that the domain size in which a derivative is defined decreases for ascending order q. The discrete Heaviside $H_g^{(q)}$ can be defined as follows

$$H_g^{(q)}(\mathbf{x}) = \begin{cases} 0 & \text{if } \|\mathbf{x} - \mathbf{x}_\Gamma\|_2 \geq q\Delta_m \text{ and } \mathbf{x} \in \Omega_l \\ 1 & \text{if } \|\mathbf{x} - \mathbf{x}_\Gamma\|_2 < q\Delta_m \text{ and } \mathbf{x} \in \Omega_l \\ 1 & \text{if } \mathbf{x} \in \Omega_g \end{cases} \tag{B.7}$$

Another way around proposed in [198] for quadratic extrapolation is the use of $H_g^{(0)}$ for the 3 PDEs. The second order normal derivative $\Phi^{(2)}$ is retrieved using full tensor-projection formulation instead of Eq. (B.6) which only implies Φ values

$$\Phi^{(2)} = \mathbf{n} \cdot \nabla \left(\mathbf{n} \nabla \Phi \right) \tag{B.8}$$

one-sided second order finite differences are used for first and second order derivatives.
This form has also been discussed in [127]. Both formulations Eqs. (B.6) and (B.8) are equivalent in the continuous sense. However, authors argued that the use of the full tensor-projection formulation avoids additional errors in the case where normal $\mathbf{n}$ is obtained from numerical differentiation of c.
Finally, the spatial discretization of Eq. (B.5) uses an upwind scheme based on the direction of the normal. The initial work of Aslam [9] used a first order upwind scheme while in other works such as [127] second order upwind scheme is used. Note that this choice can affect the overall accuracy of the procedure. In deed, McCaslin et al. [127] noticed that using a first order upwind discretization prevents a third order convergence rate with the mesh size for quadratic and higher extrapolation order. For the same reason, a second order upwind discretization does not allow fourth order accuracy using a cubic extrapolation.
For the temporal discretization one can use Euler or RK schemes for higher order temporal integration. This choice will impact the convergence speed of the method to the steady state without modifying the accuracy of the final extrapolated field.
$\Delta\tau$ can be chosen as for the reinitiailization step of Eq. (3.37).
Finally, the equations could be solved until a threshold is reached for the steady state, but most of the time, the quantity is only needed in a narrow band around the interface, and only few timesteps are required. This allows to enhance efficiency without losing accuracy close to the interface.

B.1.2 FMM approach

The PDE approach can suffer from efficiency issues if the number of iterations to perform is high to reach the steady state. Another way around which was constructed for efficient computation of signed distance is the FMM.
As for the distance computation presented in Section 3.2.1, it is also possible to consider the static form of the Eikonal equation with a speed ∇s

$$\nabla s \cdot \nabla \phi = 0 \tag{B.9}$$

The equation Eq. (B.9) was first solved in [197] to build an extension velocity defined initially at the front for the transport of the signed distance. This methodology was built to avoid the reinitialization step for SLS by transporting the signed distance directly with the extension velocity field well-defined everywhere in the domain.
The idea has been extended to constant extrapolation of quantities only defined at the interface in [80] by solving Eq. (B.9). This procedure is called "redistribution" of the given quantity.
In [127], Eq. (B.9) is extended to higher order extrapolation by considering the static form of Eq. (B.5) which is the Eikonal equation with a speed $\nabla s = \nabla \Phi^{(q)} / \Phi^{(q+1)}$

$$\nabla \Phi^{(q)} \cdot \nabla \phi = \Phi^{(q+1)} \tag{B.10}$$

this equation can be solved using the same front propagation methodology originally developed to solve Eq. (3.47). Note that the phase indicator χ^g is taken into account by only

distributing the quantity $\Phi^{(q)}$ in the gas domain from the front.
The discrete form of Eq. (B.10) gives the following equation in 2D

$$\mathcal{S}_x^- \left(\frac{\partial \Phi^{(q)}}{\partial x}^- \frac{\partial \phi}{\partial x}^- \right) + \mathcal{S}_x^+ \left(\frac{\partial \Phi^{(q)}}{\partial x}^+ \frac{\partial \phi}{\partial x}^+ \right) + \mathcal{S}_y^- \left(\frac{\partial \Phi^{(q)}}{\partial y}^- \frac{\partial \phi}{\partial y}^- \right) + \mathcal{S}_y^+ \left(\frac{\partial \Phi^{(q)}}{\partial y}^+ \frac{\partial \phi}{\partial y}^+ \right) = \Phi^{(q+1)} \tag{B.11}$$

with the switches terms $\mathcal{S}_x^{\pm}$ defined as

$$\mathcal{S}_x^+ = \begin{cases} 1 & \text{if } \max\left(\frac{\partial \phi}{\partial x}^-, -\frac{\partial \phi}{\partial x}^+, 0\right) = -\frac{\partial \phi}{\partial x}^+ \\ 0 & \text{otherwise} \end{cases} \tag{B.12}$$

$$\mathcal{S}_x^- = \begin{cases} 1 & \text{if } \max\left(\frac{\partial \phi}{\partial x}^-, -\frac{\partial \phi}{\partial x}^+, 0\right) = \frac{\partial \phi}{\partial x}^- \\ 0 & \text{otherwise} \end{cases} \tag{B.13}$$

again, the derivatives $\frac{\partial}{\partial x}^-$ and $\frac{\partial}{\partial x}^+$ are approximated with left and right biased first order finite differences.
A complete study of both PDE and FMM based extrapolations is provided in [127] where FMM shows huge efficiency improvements compared to the PDE approach even in parallel implementations. However, the computation is limited to second order accuracy because of the first order accuracy of the derivatives used to solve Eq. (B.11).

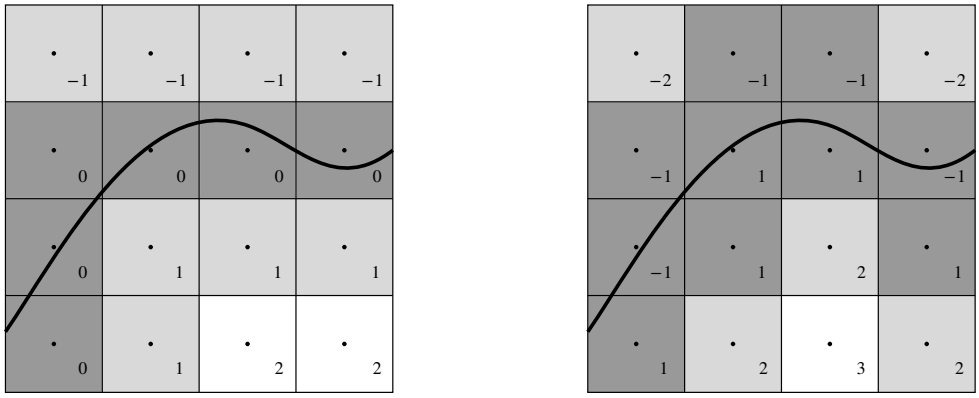
B.1.3 Neighbour propagation approach

A last approach, used by Kunkelmann [102] to pass data (here temperature gradients) from pure cells of a given phase to the interface cells in a VOF framework.
First, a narrow band is defined by tagging the closest cells to the interface. An example is given in Fig. B.2 where the number in each cell represents the band layer.
If the cell belongs to Ω_g, then the band layer tag is a negative integer while it is positive in Ω_l. For a VOF representation, the index can be zero in the case of mixed cells. This is because in such case, the cell does not belong to any phase.
From this tagging, an interface cell with $n_b = 0$ is a receiving cell which requires passing informations. This is done by a simple average of data from any surrounding cells with $n_b = 1$ for the case of liquid quantities. For a given cell $\mathcal{C}_i$, this gives

$$\Phi_i = \frac{\sum_{j \in I^1_{\Omega_l}} \Phi_j}{\sum_{j \in I^1_{\Omega_l}} 1} \tag{B.14}$$

with $I^1_{\Omega_l}$ the set of indices of neighbours cells belonging to the liquid first band ($n_b = 1$).
In [123], the receiving cell retrieves the quantity using weighted average of neighbouring cells from a 5×5 stencil. For a given receiving cell $\mathcal{C}_i$, this gives

$$\Phi_i = \sum_{j \in I^2_{\Omega_l}} W_j \Phi_j \tag{B.15}$$



(a) *VOF* **(b)** *Level Set*

Figure B.2: *Narrow band tagging for VOF and LS representation. The numbers correspond to the cell narrow band layer. The cell colours correspond to the different narrow band layers. Dark grey cells belong to* Ω_Γ.

with $I^1_{\Omega_l}$ the set of indices of cells belonging to the liquid first and second bands ($n_b = 1$ or $n_b = 2$) of the 5×5 stencil and the weights W_j a function of the distance between $\mathcal{C}_i$ and $\mathcal{C}_j$ and the collinearity between the normal in $\mathcal{C}_j$ and the direction of the displacement line between $\mathcal{C}_i$ and $\mathcal{C}_j$. This gives higher importance to collinear cells with the lowest distance to the receiving cell.
Finally, a quite similar approach is used in an unstructured LS framework [187] where the passing of data from cells with $n_b = 2$ to cells with $n_b = 1$ is performed using a second order Taylor expansion.

B.2 Study of PDE approach accuracy

One important feature of a phase-change solver is the extension of phase information in the other phase through normal extrapolation. This is done either by using a VOF or a SLS representation. As the method requires to solve successive PDEs, the temporal and spatial discretization are investigated. Then the impact of the normal computation on the normal extrapolation accuracy is discussed as VOF and SLS does not provide the same normal accuracy. Finally, linear extrapolation with imposed boundary conditions presented in Section 8.3.2 are compared to the general linear extrapolation.
The extrapolation is performed from Ω_l to Ω_g delimited by the circle of radius $R = 0.2$ centered at $(0.5, 0.5)$ in a domain $[1 \times 1]$. The target scalar field Φ is defined as in [30]

$$\Phi = \exp\left(4\left(x - 0.5\right)\left(y - 0.5\right)\right) \quad . \tag{B.16}$$

In the following, only the extrapolation in a narrowband of five cells in the interface vicinity is considered as it is the only relevant values in a solver implementation. Indeed, scalar values are not needed everywhere and are just used as ghost cells close to the interface to

avoid using erroneous values on the other side of the interface.
The error is then computed only in the narrow band as

$$L_2(\Phi) = \sqrt{\frac{1}{N_{\mathcal{B}}} \sum_{i=1}^{N_{\mathcal{B}}} (\Phi_{i,exact} - \Phi_i)^2} \quad , \tag{B.17}$$

$$L_\infty(\Phi) = \max_i |\Phi_{i,exact} - \Phi_i| \quad , \tag{B.18}$$

with $N_{\mathcal{B}}$ the number of points in the narrow band.

B.2.1 Time integration

For the time integration of Eq. (B.5), the pseudo-time step $\Delta\tau$ and number of iteration N_{ite} is required in order to obtain a steady state in the narrow band around the interface. The pseudo-time step has to respect the CFL condition

$$\Delta\tau < \frac{\Delta x}{\sum_i^{N_{dim}} n_i} \quad . \tag{B.19}$$

For safety, it is simply taken as

$$\Delta\tau = \frac{\Delta x}{N_{dim}} \quad , \tag{B.20}$$

with this time step restriction, the information propagating to a narrow band of length $5\Delta x$ requires approximately $10\Delta\tau$ to reach the steady state. In practice, depending on the threshold, the residual requires more than 10 iterations to converge to a reasonable threshold.
The time integration is performed with a SSP-RK2 scheme.
In Fig. B.3, the residual of $r(\Phi) = L_1\left(\Phi^{n+1} - \Phi^n\right)$ in the narrow band is computed with respect to the number of iterations and stops at $r(\Phi) = 10^{-12}$. It can be observed that for $N_{ite} = 20$, the residual is already very low. However, for very meticulous convergence studies such as 1D test cases, the choice is to reach the threshold $r(\Phi) = 10^{-12}$. In the more complete test cases, $N_{ite} = 20$ is used as it has no impact on the results.

B.2.2 Spatial discretization of the PDE

Now that the time integration has been defined, the discretization of Eq. (B.5) is evaluated as it could have an impact on the accuracy of the results. In this study, three discretizations of the gradient are considered:

- A first order upwind scheme for all successive (UW1)
- A second order upwind scheme for all successive (UW2)
- A first order upwind scheme for $q = m$ and a second order upwind scheme for all order derivatives $q < m$. (UWH)

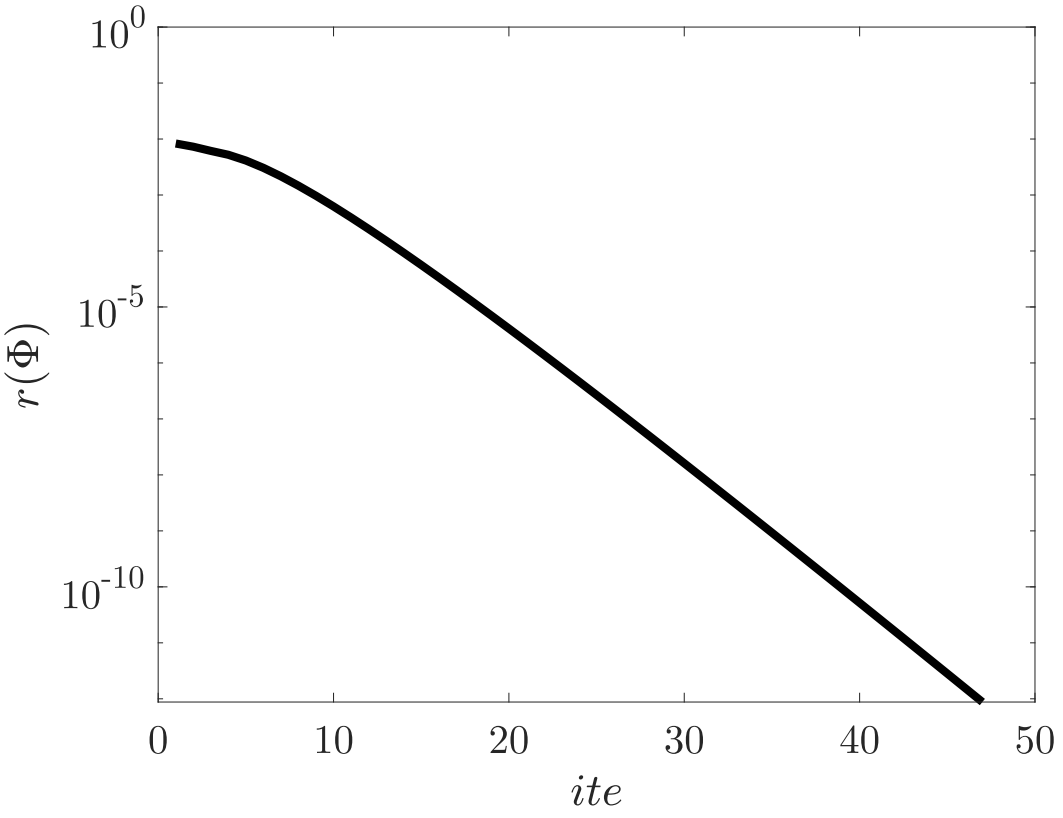
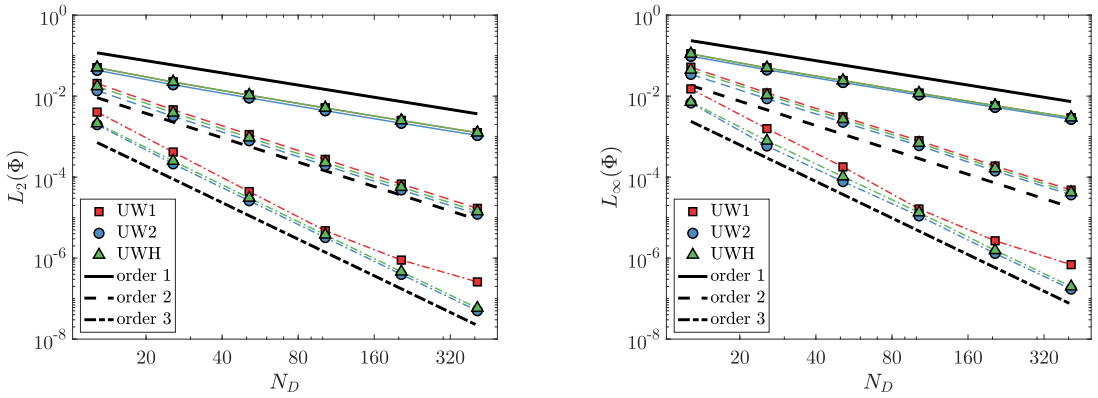


Figure B.3: *Residual convergence of* Φ *in a narrow band around the interface*



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

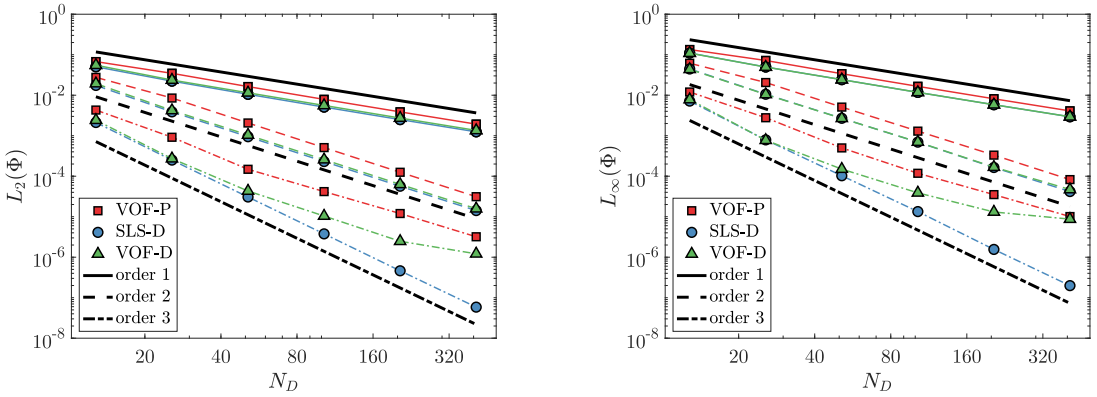
Figure B.4: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the different gradient approximations using* $m = 1$ (——)*,* $m = 2$ (- - -) *and* $m = 3$ (-···-)*.*

The convergence of $L_2(\Phi)$ and $L_\infty(\Phi)$ is presented in Fig. B.4. The use of UW1 and UW2 does not seem to affect the $m = 1$ and $m = 2$ extrapolation even if just a small improvement is observed for UWH and UW2. However, for $m = 3$, the use of UW1 does not lead to the expected third order convergence of the error. In [127], it has already been noticed that using a first order upwind discretization prevents a third order convergence rate with the mesh size for quadratic and higher extrapolation order.
A reason for using UWH instead of UW2 is to prevent any extrema creation on $\Phi^{(m)}$. For example, a constant field subject to some spurious errors close to the interface could lead to a non-uniform field in the extrapolated area because of the larger stencil.
For the following studies and in the solver, the UWH is used in order to have both accuracy and robustness of the extrapolation method.

B.2.3 Normal computation impact

The normal is not evaluated in the same manner using VOF or SLS representation of the interface. For SLS, this is obtained by simple differentiation of ϕ, however, for VOF, the normal is only available at the interface using ELVIRA. The normal used in the PDE procedure needs to be defined in the narrow band and for VOF two different methods are considered. In total, the three following methods are compared:

- The simple neighbour propagation presented in Appendix B.1.3 (VOF-P)
- The differentiation of the signed distance ϕ (SLS-D)
- The differentiation of a signed distance function reconstructed from the isocontour $f^{-1}(0.5)$ as in [192; 27]. (VOF-D)



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

Figure B.5: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the different normal approximations using* $m = 1$ (——), $m = 2$ (- - -) *and* $m = 3$ (-···-).

In Fig. B.5 are presented $L_2(\Phi)$ and $L_\infty(\Phi)$ for different normal approximations for $m = 1$ to $m = 3$. It is interesting to notice that the normal approximation has an impact on extrapolation accuracy. The very basic method VOF-P does not lead to a very accurate normal field and then the accuracy of the extrapolated field is also reduced both in $L_2(\Phi)$

and $L_\infty(\Phi)$. This discrepancy is increasing with the order m. This normal computation can be improved by using the VOF-D procedure. It is able to retrieve the same results as the SLS normal for the coarse meshes. However this is not the case for $m = 3$ where the convergence rate is decreasing. This convergence saturation can be explained by the dependence on the mesh of the DVOF procedure which prevents convergence for high resolution.
In the following, the VOF-D procedure will not be used as it does not lead to a significant accuracy improvement regarding the additional computational cost of computing the signed distance.

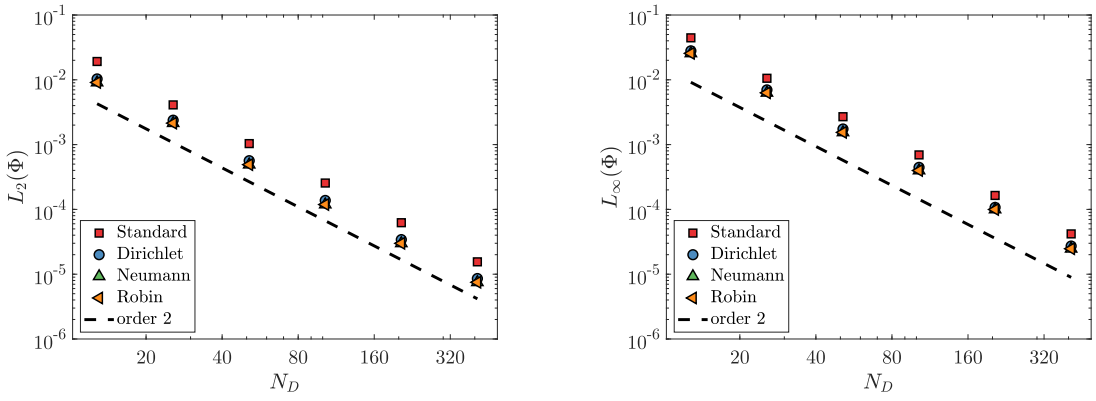
B.2.4 Constrained extrapolation

In Section 8.3.2, a variant of the normal extrapolation is proposed to take into account a boundary condition at the interface between the liquid and the gas. It is interesting to compare the accuracy of such constrained extrapolation with the linear extrapolation considered before. Here, it is possible to apply either Dirichlet, Neumann or Robin boundary conditions by using Eq. (8.42) with

$$\Phi_\Gamma = \exp\left(4\left(x_\Gamma - 0.5\right)\left(y_\Gamma - 0.5\right)\right) \quad , \tag{B.21}$$

$$\Phi_\Gamma^{(1)} = 4\left((x_\Gamma - 0.5)n_x + (y_\Gamma - 0.5)n_y\right)\exp\left(4\left(x_\Gamma - 0.5\right)\left(y_\Gamma - 0.5\right)\right) \quad . \tag{B.22}$$

This study is limited to the linear version of the algorithm which remains straightforward to implement, higher order implementations requires additional interpolation steps in the liquid domain.



(a) *Convergence of* $L_2(\Phi)$ **(b)** *Convergence of* $L_\infty(\Phi)$

Figure B.6: *Mesh convergence of* $L_2(\Phi)$ *and* $L_\infty(\Phi)$ *for the different constraint at the boundary*

From the results shown in Fig. B.6, the constrained extrapolation is more accurate than the standard procedure (error divided by two) with only few differences between the nature of the boundary condition. This is expected as the computation of the gradient at the interface is computed with higher accuracy and then the forcing term in the $q = 1$ PDE is

leading to a better linear approximation of Φ.
However, constrained extrapolation is limited to second order (for reasonable implementation effort) and requires computation of boundary conditions, which is not always available. Moreover, the implementation is not straightforward in a VOF fashion.
In the solver, the standard procedure is used to keep a general way of extrapolating quantities regardless the interface capturing method or the boundary conditions at the interface.

B.2.5 Method choice recap for extrapolation

Table B.1 summarizes the different numerical choices for the extrapolation procedure. This work uses the UWH method for spatial discretization as it is accurate and stable. The time integration is performed with a SSP-RK2 scheme with the constraint Eq. (B.20) on the pseudo-timestep. Only 20 iterations are performed instead of reaching a target convergence threshold.
Even if the VOF-D method provides more accurate extrapolated quantities, a good trade-off between accuracy and efficiency is achieved by using the VOF-P method instead.

Interface capturing method	VOF	SLS
Time integration	RK2-SSP	RK2-SSP
Number of iteration	20	20
Spatial discretization	UWH	UWH
Normal computation	VOF-P	SLS-D

Table B.1: *Summary of computation choices for the extrapolation procedure*

In order to use standard finite differences using uniform stencils, the scalar need to be defined everywhere. This is done by the normal extrapolation investigated in Appendix B.2. The normal gradient is then simply evaluated using Eq. (8.32). In the present study, the constant extrapolation is not suitable for gradient computation as it is not a consistent choice leading to zero normal gradient. Then, only linear and quadratic extrapolations are considered.

B.3 Solver discretization details

In the solver, the PDEs are discretized as follows

$$\begin{aligned}
\frac{\Phi_l^{(2),n+1} - \Phi_l^{(2),n}}{\Delta\tau} &= -H_g^{(2)} \left(n_x \nabla^x_{c,UW1} \Phi_l^{(2),n} + n_y \nabla^y_{c,UW1} \Phi_l^{(2),n} - \Phi_l^{(3)} \right) \\
\frac{\Phi_l^{(1),n+1} - \Phi_l^{(1),n}}{\Delta\tau} &= -H_g^{(1)} \left(n_x \nabla^x_{c,UW2} \Phi_l^{(1),n} + n_y \nabla^y_{c,UW2} \Phi_l^{(1),n} - \Phi_l^{(2)} \right) \\
\frac{\Phi_l^{n+1} - \Phi_l^{n}}{\Delta\tau} &= -H_g^{(0)} \left(n_x \nabla^x_{c,UW2} \Phi_l^{n} + n_y \nabla^y_{c,UW2} \Phi_l^{n} - \Phi_l^{(1)} \right) \quad ,
\end{aligned} \tag{B.23}$$

and

$$\begin{aligned}
\frac{\Phi_g^{(2),n+1}-\Phi_g^{(2),n}}{\Delta\tau} &= H_l^{(2)}\left(-n_x\nabla^x_{c,UW1}\Phi_g^{(2),n}+n_y\nabla^y_{c,UW1}\Phi_g^{(2),n}+\Phi_g^{(3)}\right)\\
\frac{\Phi_g^{(1),n+1}-\Phi_g^{(1),n}}{\Delta\tau} &= H_l^{(1)}\left(n_x\nabla^x_{c,UW2}\Phi_g^{(1),n}+n_y\nabla^y_{c,UW2}\Phi_g^{(1),n}+\Phi_g^{(2)}\right)\\
\frac{\Phi_g^{n+1}-\Phi_g^{n}}{\Delta\tau} &= H_l^{(0)}\left(n_x\nabla^x_{c,UW2}\Phi_g^{n}+n_y\nabla^y_{c,UW2}\Phi_g^{n}+\Phi_g^{(1)}\right) \quad ,
\end{aligned} \tag{B.24}$$

with $\Delta\tau$ defined by Eq. (B.20), $H_l^{(q)}$ and $H_g^{(q)}$ defined by Eq. (B.7).
The first order upwind gradient operators $\nabla^x_{c,UW1}$, $\nabla^y_{c,UW1}$ are defined (here for $\nabla^x_{c,UW1}\Phi\Big|_{i-\frac{1}{2},j}$)

$$\nabla^x_{c,UW1}\Phi\big|_{i-\frac{1}{2},j} = \begin{cases} \frac{\Phi_i-\Phi_{i-1}}{\Delta x} & \text{if } n_x>0 \\ \frac{\Phi_{i+1}-\Phi_i}{\Delta x} & \text{otherwise} \end{cases} \quad , \tag{B.25}$$

and the second order upwind gradient operators $\nabla^x_{c,UW2}$ and $\nabla^y_{c,UW2}$ are defined (here for $\nabla^x_{c,UW2}\Phi\Big|_{i-\frac{1}{2},j}$)

$$\nabla^x_{c,UW2}\Phi\big|_{i-\frac{1}{2},j} = \begin{cases} \frac{3\Phi_i-4\Phi_{i-1}+\Phi_{i-2}}{2\Delta x} & \text{if } n_x>0 \\ \frac{\Phi i+2-4\Phi_{i+1}+3\Phi_i}{2\Delta x} & \text{otherwise} \end{cases} \quad . \tag{B.26}$$

Note that the sign of the normal used in Eq. (B.24) is also the liquid to gas normal and some signs have been modified accordingly to extrapolate from gas to liquid. For LE, only the equations for $T_l^{(1),n+1}$ and T_l^{n+1} need to be solved.
As normal gradients $T_{\Gamma,l}^{(1)}$ and $T_{\Gamma,g}^{(1)}$ are only available in interface cells Ω_Γ, the evaporation $\dot{m}$ is only defined in Ω_Γ.

Appendix C

Resumé étendu en français

Introduction

La simulation des écoulements diphasiques évaporants est un sujet en plein essor avec l'émergence de multiples solveurs et méthodes numériques associées. C'est un sujet complexe en raison des discontinuités de l'écoulement à l'interface qui nécessitent des méthodes numériques adaptées.

Dans la littérature, les méthodes employées pour résoudre les écoulements diphasiques peuvent être classifiées en deux manières de représenter l'interface:

1. Les méthodes d'interface diffuse (DIM) qui définissent l'interface comme une région où la variation des propriétés fluides est rapide mais reste continue.
   Dans ce cas, l'interface n'a pas de position explicitement définie. Ces méthodes sont généralement utilisées pour étudier des écoulements compressibles puisqu'elles se basent sur une thermodynamique enrichie pour prendre en compte la nature diphasique de l'écoulement.
2. Les méthodes d'interface raide (SIM) qui utilisent un interface infiniment fin. Cet interface est donc une surface mobile avec une position bien définie. Les propriétés fluides sont alors purement discontinues à l'interface et l'interface nécessite d'être localisé précisément pour y imposer correctement les sauts.

Dans ce travail, l'écoulement est considéré incompressible. Il est intéressant d'utiliser une approche SIM dans ce contexte puisqu'elle requiert généralement moins de points pour représenter l'interface. De plus l'hypothèse d'interface infiniment fin est pertinent dans les configurations ciblées.
Dans ces approches SIM, les méthodes Lagrangiennes de type Front-Tracking ne sont pas considérées dans ce travail pour des raisons pratiques d'intégration de ce type d'algorithme dans le solveur développé au laboratoire EM2C.

Parmi les méthodes Eulériennes raides, plusieurs stratégies ont montré leur capacité à reproduire la physique des écoulements diphasiques sans changement de phase, que ce soit en Volume-of-Fluid (VOF) ou en Level-Set (LS). Dans ce contexte, la littérature est abondante

et les méthodes numériques associées relativement matures.
Les problématiques liées à la reconstruction précise de la courbure, essentielle à la prise en compte des forces de tension de surface ont été explorées en détails et permettent la simulation numérique précise d'écoulement capillaires. De plus, le manque de robustesse lié au fort rapport de masse volumique entre le liquide et le gaz peut être comblé par un transport cohérent entre la masse et la quantité de mouvement qui rend possible la simulation numérique de configurations extrêmement exigeantes comme des sprays atomisants.

Avec le changement de phase, les méthodes numériques gagnent encore en complexité pour prendre en considération les échanges de masse et de chaleur entre les phases. Dans ce cadre, peu de contributions existent et aucun consensus ne ressort de la littérature sur la meilleure stratégie à adopter.
Des études spécifiques sont alors nécessaires pour atteindre le même niveau de compréhension que pour les écoulements sans évaporation.

Ainsi, ce travail propose de clarifier les méthodes numériques existantes pour répondre aux difficultés de la simulation numérique d'écoulement diphasique à changement de phase. Cela implique d'étudier les méthodes Eulériennes raides pour la capture d'interface, le couplage entre le transport de l'interface et l'écoulement, ainsi que l'intégration du changement de phase dans le solveur. Cette étude est menée dans un solveur basé sur une discrétisation de type volumes finis sur des maillages cartésiens dans lequel les méthodes sont implémentées et comparées.

Partie I

La première partie du manuscrit traite de la dérivation rigoureuse des équations de conservation dans le cas d'écoulements diphasiques à changement de phase.

Tout d'abord, le modèle physique utilisé dans ce travail pour les fluides ainsi que la dynamique de l'interface sont détaillés. Les équations obtenues s'appuient sur les principes généraux de conservation de masse, quantité de mouvement et d'énergie appliqués à des écoulements incompressibles, avec des propriétés fluides constantes pour chaque phase prise séparément.
Pour la modélisation du changement de phase, l'écoulement diphasique est composé d'une phase liquide mono-composant pure et d'une phase gazeuse bi-composant comprenant la vapeur de l'espèce condensable et un gaz inerte. Il est donc nécessaire de résoudre une équation supplémentaire correspondant au transport de fraction massique de l'espèce condensable pour compléter le système.
Ces hypothèses de travail mènent à un système d'équations de conservation valable dans les fluides pris séparément ainsi qu'à des conditions de saut découlant directement d'un équilibre entre les phases dans un volume de contrôle contenant l'interface.

Ces discontinuités impliquent des quantités à fermer : la force de tension de surface ainsi

que le taux d'évaporation. Alors que la force de tension de surface peut se calculer à partir des propriétés topologiques de l'interface, le taux d'évaporation s'obtient avec la discontinuité de flux d'énergie ou de fraction d'espèce à l'interface.
Ces équations de conservation ne sont pas suffisantes pour décrire l'écoulement diphasique puisqu'elles ne s'appliquent qu'aux fluides pris séparément en laissant de côté les contributions liées à la présence de l'interface et aux échanges entre ces deux fluides.

De manière à obtenir des équations capables de décrire l'écoulement diphasique dans sa globalité, une représentation explicite de l'interface à travers une fonction indicatrice peut être utilisée. La fonction indicatrice est associée à un fluide et donne sa position pour tous points de l'espace en tout temps. Elle est intrinsèquement discontinue et son transport devient alors extrêmement difficile numériquement.
C'est pourquoi l'utilisation de discrétisations conventionnelles pour transporter cette grandeur mènerait à la diffusion ou la dispersion de la discontinuité ce qui met en péril la représentation raide et précise de l'interface.

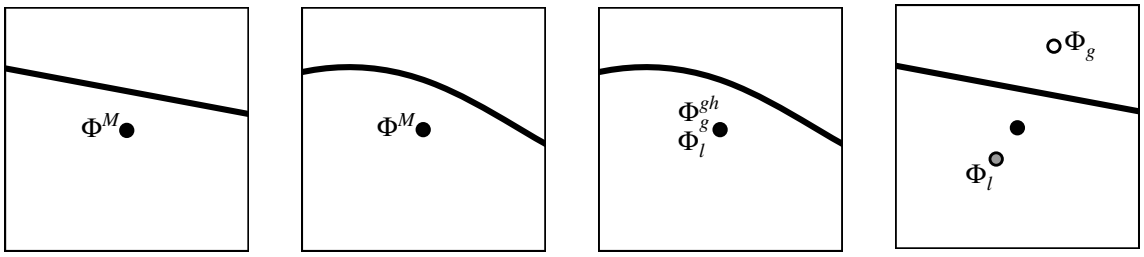
De ce constat, une première approche est proposée en se basant sur une discrétisation volumes finis de la fonction indicatrice et de son transport. Ce choix de représentation numérique de la fonction indicatrice s'apparente aux méthodes VOF dans lesquelles la fraction volumique de liquide est transportée. En partant de ce formalisme, la dérivation des formulations VOF "two-fluid" puis "one-fluid" est obtenue par intégration des équations de conservation.
Ces formulations impliquent le transport de la fraction volumique et comportent encore la fonction indicatrice dans l'évaluation des flux. De manière à évaluer précisément ces flux, une représentation explicite de l'interface est nécessaire, ce qui constitue une contrainte importante sur les méthodes numériques associées au transport des quantités fluides.

Pour contourner ce problème, une autre méthodologie est proposée, pour laquelle la fonction indicatrice est substituée par une fonction Level-Set régulière. Cette substitution retire une partie de la cohérence entre les équations de conservation et le transport de l'interface mais permet de gagner des degrés de liberté sur les méthodes numériques associées.
La raison de la perte de cohérence entre fonction indicatrice et formulation LS est détaillée et des solutions de la littérature sont proposées pour remédier au problème sans le résoudre totalement.
Ensuite, la méthodologie à suivre pour une cohérence complète dans la formulation Level-Set est proposée en se basant sur la définition d'un opérateur de moyenne plutôt que sur une substitution des fonctions transportées. Bien que peu explorée, cette méthodologie représente une réelle piste de recherche pour la dérivation mathématique des écoulements diphasiques.

À partir de ces deux formalismes, trois descriptions des quantités diphasiques sont retenues pour la suite des travaux, illustrées en figure C.1

- La Whole Domain Formulation (WDF) basée sur les formulations one-fluid VOF ou LS pour lesquelles une seule équation décrit le transport d'une quantité diphasique

moyennée par la phase dans un volume de contrôle.
- La Jump Condition Formulation (JCF) découlant de la formulation one-fluid LS pour laquelle la quantité diphasique est associé à une seule phase dans un volume de contrôle.
- La Two-Fluid Formulation (TFF) qui transporte simultanément la quantité liquide et gazeuse dans un volume de contrôle.



(a) *WDF avec VOF* **(b)** *WDF avec LS* **(c)** *JCF* **(d)** *TFF*

Figure C.1: *Illustration des différentes formulations pour un scalaire Φ dans un volume de contrôle. Le point noir correspond au centre de la cellule alors que les point gris et blancs sont les barycentres de la phase liquide et gazeuse respectivement.*

Alors que la WDF et la JCF nécessitent une reconstruction supplémentaire pour retrouver les quantités liées aux phases, la TFF transporte toutes les informations nécessaires directement.
Ces descriptions de l'écoulement sont réutilisées tout au long du manuscrit pour expliquer et clarifier les méthodes numériques associées à la résolution d'écoulements diphasiques. Cette étape, bien que fastidieuse, est indispensable au développement de méthodes numériques cohérentes puisqu'elle conditionne le sens des quantités transportées.

Partie II

Une fois le cadre physique et mathématique clairement posé, la deuxième partie détaille les méthodes numériques associées à la capture de l'interface.

Pour commencer, un état de l'art étendu des méthodes VOF et Level-Set est présenté de manière à sélectionner les méthodes les plus prometteuses pour représenter des écoulements évaporant.

Les challenges d'une méthode VOF résident dans la reconstruction explicite de l'interface dans un volume de contrôle. Si une représentation PLIC est considérée, la précision de la reconstruction dépendra directement du calcul de la normale à l'interface et sera au mieux à l'ordre 2.
L'autre point d'attention d'une méthode VOF réside dans l'évaluation des flux à partir de la géométrie de l'interface qui nécessite des algorithmes particuliers pour conserver discrètement la masse et éviter les undershoots ou overshoots de fraction volumique.

Pour la Level-Set, un premier choix de fonction régulière est la distance minimale signée à l'interface menant à une méthode standard LS (SLS). Le transport associé à cette fonction peut utiliser des méthodes d'ordre élevé telles que les schémas WENO5, plus précis que les flux géométriques utilisés en VOF.
Cependant, la nature de l'écoulement transportant cette fonction peut entraîner de sévères déviations par rapport à une fonction distance. Il est alors très compliqué de calculer des grandeurs topologiques précises en se basant sur cette fonction distance bruitée.
La principale particularité cette méthode est la nécessité de résoudre une équation supplémentaire au transport pour retrouver une fonction distance correcte à la fin de l'intégration temporelle. De plus, cette méthode souffre de problèmes de conservation de la masse intrinsèque à sa formulation et son manque de cohérence avec la fonction indicatrice décrite dans le deuxième chapitre.

Pour remédier à ce problème de conservation, une autre classe de méthode LS se basant sur une fonction Heaviside étalée est également détaillée, nommée l'accurate conservative LS (ACLS). Cette méthode permet de conserver exactement la fonction Heaviside, ce qui ne garantit pas la conservation de la masse mais l'améliore.
Pour les mêmes raisons qui bruitent la fonction distance dans le cas de la SLS, une étape de réinitialisation est également nécessaire après le transport pour contrôler l'épaisseur d'étalement de l'Heaviside transporté. Différentes méthodes de réinitialisation sont alors comparées pour définir la plus adaptée à la description d'écoulements diphasiques évaporants.

Enfin, une dernière méthode consiste à coupler une méthode LS et une méthode VOF de manière à avoir les propriétés de conservation du VOF tout en gardant une représentation plus précise de l'interface. En réalité, cette dernière méthode résulte en un équilibre entre propriétés de conservation et précision de transport suivant la façon de coupler LS et VOF. Les différents types de couplage ainsi que leurs intérêts et limites sont détaillés résultant en un choix de méthode CLSVOF pour la suite du travail.

Une fois que ces quatre méthodes ont été sélectionnées en se basant sur l'état de l'art étendu, leur implémentation est détaillée. Ces méthodes sont ensuite comparées sur des cas de transport avec champ de vitesse imposé. Les critères principaux sont les propriétés de conservation de masse, la précision du transport ainsi que le temps de calcul.
En guise d'exemple, le cas du vortex-in-a-box est fourni en figure C.2 où le transport d'un interface dans un champ de vitesse déformant est étudié pour les quatre méthodes de capture sélectionnées.
Dans ce cadre, même si toutes les méthodes présentent des résultats cohérents et convergents en maillage, le CLSVOF semble être la méthode la plus précise grâce à ses propriétés de conservation. De plus, sa polyvalence pourrait être un véritable atout pour des cas plus complexes.

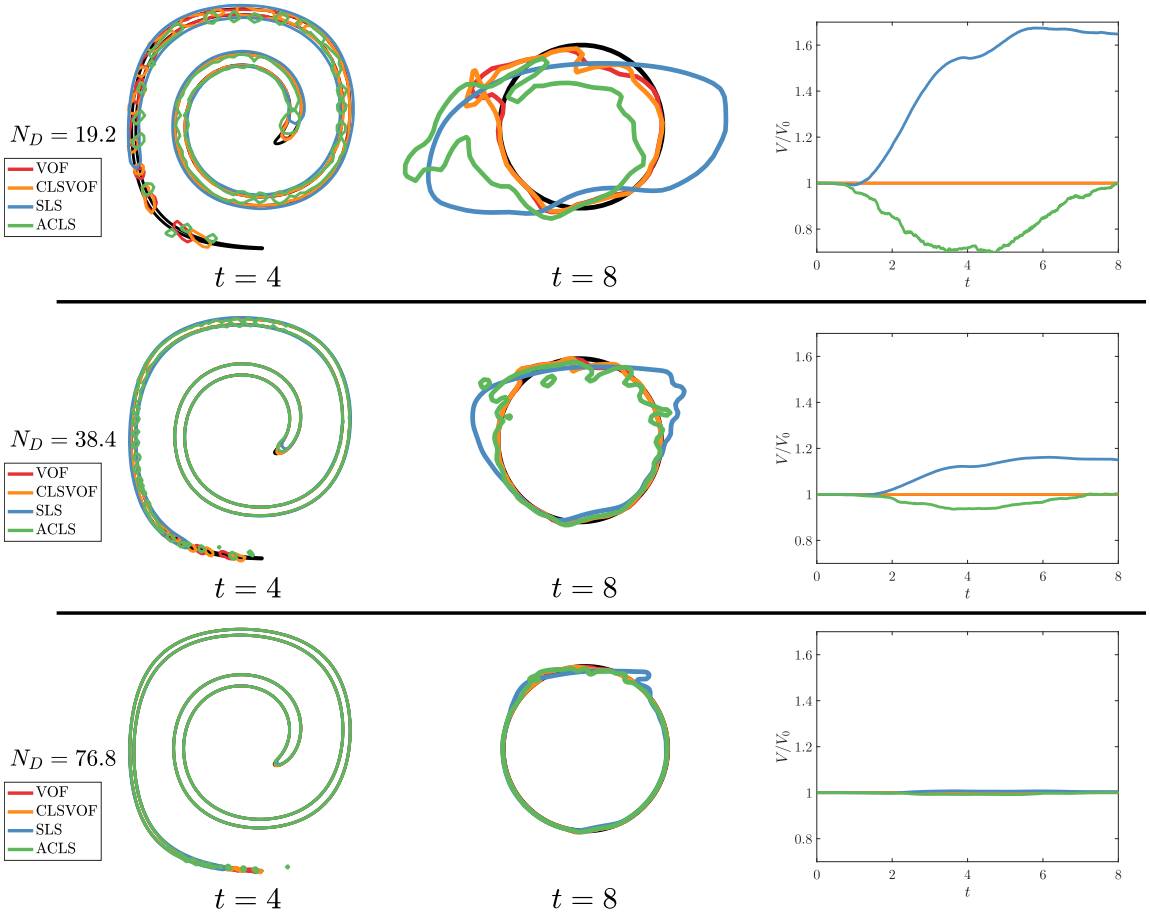


Figure C.2: *Forme du vortex-in-a-box à $t = 4$ et $t = 8$ avec l'évolution temporelle de la masse.*

Partie III

La troisième partie du manuscrit s'intéresse au couplage entre une méthode de capture d'interface et un solveur d'écoulement incompressible.

Dans un premier temps, les problèmatiques associées au couplage d'une méthode de capture d'interface avec un solveur incompressible sont détaillées.
Plus spécifiquement, le traitement des discontinuités à l'interface nécessite des méthodes numériques appropriées pour :

1. Le calcul de la courbure, essentielle à la prise en compte des forces de tension de surface.
2. La gestion du saut de masse volumique entre les deux phases, qui peut poser des problèmes de robustesse et de conservation de la quantité de mouvement pour des rapports de masse volumique élevés.
3. La prise en compte du saut de viscosité qui joue un rôle important dans la dissipation d'énergie cinétique proche de l'interface.

Ces problématiques numériques ont été largement explorées pendant ces dernières décennies et le choix des méthodes associées se base sur un état de l'art étendu des solutions disponibles dans la littérature.

L'étude présentée dans ce manuscrit se base sur une approche one-field de la vitesse qui est définie de manière continue partout dans le domaine (cela correspond aux formulations WDF et JCF présentées sur la figure C.1).
Avec cette approche, les équations de conservation de masse et de quantité de mouvement sont résolues simultanément par le biais d'une méthode de projection : la vitesse est tout d'abord prédite sans prendre en considération les effets de pression, puis corrigée par une pression obtenue en imposant la condition de divergence nulle propre aux écoulements incompressibles.
Une autre particularité de l'approche considérée dans ce travail est l'arrangement "staggered" des quantités : la masse volumique, la viscosité et la pression sont définis au centre d'un volume de contrôle alors que les composantes de vitesse sont définies aux faces.

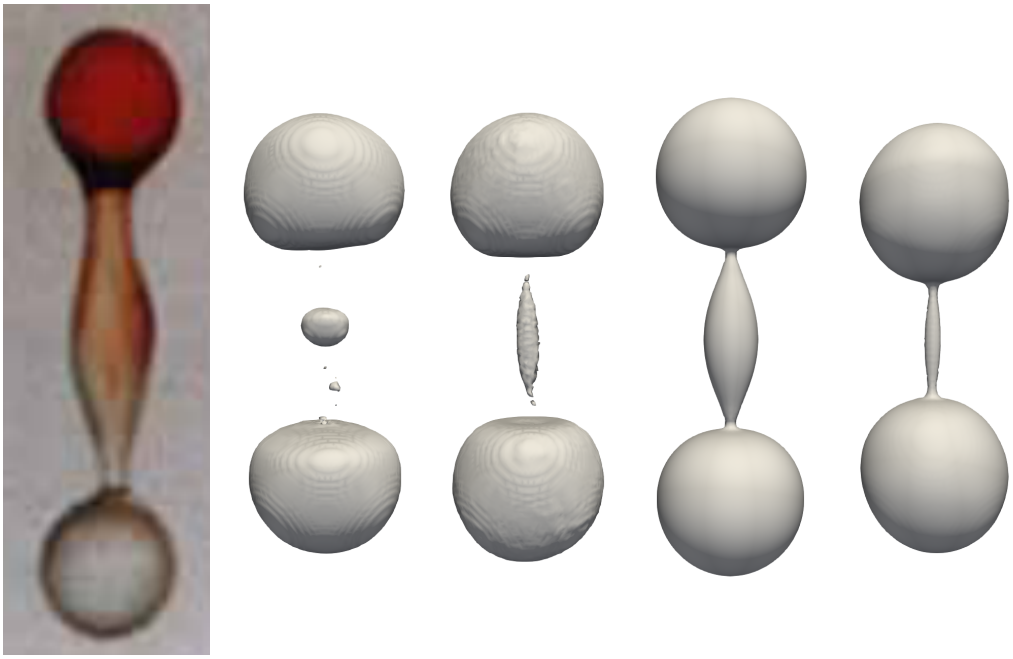
Enfin, l'originalité du travail présenté ici réside dans la définition d'un cadre unifié pouvant accueillir toutes les méthodes de capture d'interface présentées dans la partie II. Ce cadre comprend un transport précis de l'interface, une discrétisation équilibrée de la tension de surface et un transport masse-quantité de mouvement cohérent qui permet des simulations avec un rapport de densité élevé et des effets capillaires.
Ce cadre unifié, bien que contraignant sur les choix numériques, permet de construire une base solide pour des comparaisons équitables des méthodes de capture d'interface en enlevant tous les biais liés aux différences de résolution de l'écoulement.
Une description détaillée de l'implémentation des méthodes numériques adaptées au solveur unifié est ensuite fournie avec une attention particulière portée sur l'évaluation de la courbure et de la masse volumique qui diffère en fonction de la méthode de capture d'interface utilisée. En effet, les méthodes de capture d'interface ne fournissant pas les mêmes informations intrinsèques liées à l'interface, il est indispensable d'utiliser des méthodes adaptées à ces dernières pour obtenir des résultats de bonne qualité.

Les quatre méthodes de capture d'interface présentées précédemment sont alors comparées de nouveau sur des cas faisant intervenir tous les aspects d'un écoulement diphasique sans changement de phase : la modélisation des forces de tension de surface, la conservation de la masse et la conservation de la quantité de mouvement.
Il est important de rappeler que cette comparaison se base sur des méthodes actuelles de la littérature, dans un cadre unifié éliminant tous les biais associés à la discrétisation des équations de conservation sur des cas 2D canoniques mais aussi sur des cas 3D complexes. Ce travail permet donc de faire un état des lieux des méthodes actuellement disponibles et des les évaluer sur des cas tests appropriés.

Dans cette étude, une attention particulière est accordée aux cas avec tension superficielle qui font intervenir l'évaluation précise de la courbure. Cette étude augmente graduellement la complexité des configurations, en partant de la reconstruction de la courbure sur un cer-

cle statique jusqu'à la convection d'une goutte soumise à des forces de tension de surface. Ce dernier cas met en évidence la problématique principale des solveurs diphasiques actuels : aucune méthode de reconstruction de courbure n'est capable de converger en maillage sur un cas dynamique.

Ensuite un cas de conservation de quantité de mouvement est présenté mettant encore un fois en évidence les propriétés de conservation bien supérieures pour les méthodes VOF et CLSVOF par rapport aux méthode SLS et ACLS.



(a) Ashgriz et al. (b) VOF (c) CLSVOF (d) SLS (e) ACLS

Figure C.3: *Comparaison de la simulation de la collision de deux gouttes d'eau avec l'expérience*

Puis, des cas tests complexes de collision de gouttes ou de rupture induite par cisaillement sont étudiés. Les méthodes donnent des résultats qualitativement comparables avec l'expérience pour la collision de goutte (voir la figure C.3). Pour la rupture induite par cisaillement, les méthodes donnent des résultats différents, ce qui nécessite des études sur la capacité des méthodes à saisir le début des instabilités lorsqu'elle intervient en limite de maillage.

En résumé, l'étude menée dans cette troisième partie a permis de tirer plusieurs conclusions concernant les capacités des méthodes de capture d'interface actuelles :

- De manière générale, les quatre méthodes utilisées dans ce travail permettent de retrouver les bons comportements physiques pour des écoulements diphasiques sans changement de phase.

- Les méthodes VOF et CLSVOF conservent exactement la masse et permettent de retrouver des bonnes propriétés de conservation de quantité de mouvement.
- La méthode SLS donne facilement accès à des reconstructions de normales et de courbures robustes.
- Toutes les méthodes présentent des comportement différents en limite de maillage.

Partie IV

La dernière partie du manuscrit détaille l'intégration du changement de phase dans un solveur diphasique en utilisant la méthode VOF ou SLS.

Avec le changement de phase, les sauts d'interface doivent tenir compte des transferts de masse et de chaleur. Ces spécificités supplémentaires impactent la résolution des écoulements à plusieurs niveaux :

1. Les algorithmes développés précédemment pour résoudre l'équation de masse, de quantité de mouvement ainsi que la capture d'interface doivent prendre en compte le saut de vitesse induit par l'évaporation.
2. Le transport de nouvelles quantités est nécessaire de manière à décrire le changement de phase : l'énergie et la fraction d'espèce massique de la vapeur dans le gaz. Ces équations de transport doivent également inclure les discontinuités de flux moléculaires à l'interface.
3. Les transferts de masse et de chaleur nécessitent la connaissance du taux d'évaporation local qui doit être évalué précisément pour obtenir des simulations qui convergent en maillage.

Dans un premier temps, ces spécificités liées au changement de phase font l'objet d'un état de l'art étendu dans lequel les différentes méthodologies employées dans la littérature sont comparées dans leur paradigme et reliées au cadre unifié présenté précédemment.

Ces nouveaux défis sont ensuite étudiés numériquement afin de justifier les choix effectués dans le solveur. Une attention particulière est consacrée aux erreurs numériques dans la reconstruction des quantités d'interface, montrant l'importance des méthodes d'extrapolation et de l'évaluation des quantités à la position de l'interface. La question du choix de la fomulation choisie pour transporter les nouvelles quantités est également abordée en sélectionnant une approche TFF pour le VOF et JCF pour la SLS.

Le cadre unifié peut alors être étendu au changement de phase et les détails de discrétisation et d'implémentation des méthodes sont donnés.
Une analyse quantitative est menée sur des problèmes d'évaporation canoniques avec solutions analytiques. Les méthodes VOF et SLS montrent une convergence en maillage, mais avec des tendances opposées pour le cas d'évaporation statique : alors que la SLS surestime la régression de l'interface, le VOF la sous-estime.
Cela s'explique par les problèmes de conservation intrinsèques à la SLS qui font varier la masse numériquement alors que le VOF, donnant accès à des reconstructions moins pré-

cises, commet des erreurs plus importantes sur l'approximation du taux d'évaporation. Ces erreurs disparaissent avec le raffinement du maillage mais restent très importantes pour les maillages les plus grossiers.
Une conclusion émerge alors de ces observations : en utilisant les stratégies numériques actuelles pour la simulation d'écoulement diphasique à changement de phase, il est nécessaire d'avoir un maillage très raffiné pour obtenir des résultats exploitables. Cela constitue une limitation sérieuse de l'applicabilité de ces méthodes à des cas plus complexes tels que des distributions de gouttes.

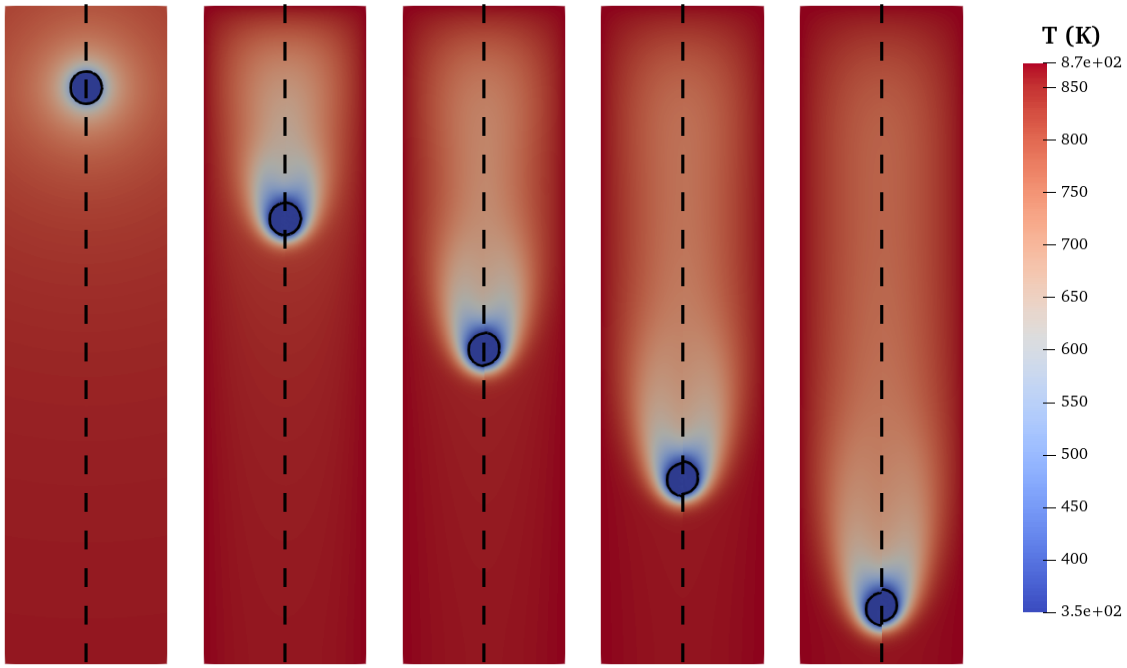


Figure C.4: *Imagesd'une goutte convectée à plusieurstemps physiques (intervalles de* 0.002 *s). La simulation VOF est à droite et la simulation SLS est à gauche.*

Finalement, le dernier cas test est une goutte convectée dans un écoulement au repos, qui implique une convection, une déformation de l'interface et une vaporisation non-homogène (voir la figure C.4). L'importance de la cohérence entre l'équation de quantité de mouvement et de continuité est mise en évidence en considérant différentes techniques de la littérature pour l'équation de continuité. Il est montré que les formulations incohérentes conduisent à une augmentation de la vitesse de la goutte. Les résultats pour la SLS suggèrent que le cadre unifié est trop contraignant, et qu'un cadre dédié doit être utilisé. Enfin, comme les équations VOF peuvent être rigoureusement dérivées des équations de conservation, il n'y a aucune ambiguïté dans l'équation de continuité discrète, et nous considérons que les résultats sont représentatifs de la physique.

Conclusion

Ce travail de thèse mène à une meilleure compréhension des méthodes numériques appliquées à la résolution des équations de conservation d'écoulements diphasiques. À travers les différentes expériences numériques menées, la complexité et la subtilité des méthodes numériques associées à la simulation d'écoulements diphasiques avec changement de phase ont été mise en évidence.
La contrainte d'un cadre unifié imposée dans ces travaux a démontré l'intime relation entre le choix de représentation de l'interface (la méthode de capture d'interface) et la formulation des équations diphasiques utilisée (la fermeture des contributions à l'interface).
Comme les méthodes VOF dérivent naturellement des formulations "two-fluid" et "one-fluid" appliquées à un cadre volumes finis, la fermeture des contributions à l'interface s'écrit directement à partir des équations de conservation.

Les mêmes dérivations "two-fluid" et "one-fluid" à partir d'une méthode Level-Set sont bien plus compliquées à obtenir et n'ont pas encore été présentées dans la littérature. À la place, des hypothèses fortes sont utilisées pour fermer les contributions à l'interface qui se basent sur des considérations numériques plus que mathématiques.
Cela donne l'illusion que les méthodes Level-Set comportent un nombre important de degrés de liberté sur les fermetures des contributions à l'interface. Cette idée est même confirmée par les cas tests conduits sans changement de phase pour lesquels toutes les méthodes de capture d'interface sont performantes et donnent des résultats des résultats physiques similaires.

Cependant, ce travail met évidence les problèmes numériques émergeant de la prise en compte du changement de phase en utilisant la formulation Level-Set. En effet, en appliquant le même cadre unifié aux méthodes VOF et SLS, les résultats dépendent fortement de la façon dont le saut de vitesse, inhérent au changement de phase, est traité dans la méthode de projection. Dans le cas de la SLS, le cadre unifié basé sur la formulation WDF ne permet pas de retrouver des résultats physiques acceptables à cause des fermetures utilisées pour les contributions à l'interface.
Il est alors clair que l'ajout du changement de phase dans un solveur diphasique introduit implicitement des contraintes numériques supplémentaires. Alors que les méthodes VOF utilisent les dérivations naturellement obtenues dans un cadre volumes finis WDF et TTF, la méthode Level-Set est plus adaptée à une formulation JCF avec des fermetures numériques adaptées.

Enfin, il est important de rappeler que les méthodes actuelles, bien que sophistiquées et rigoureusement appliquées à un solveur d'écoulement diphasique avec changement de phase, requièrent des maillages très raffinés pour obtenir des résultats exploitables. De manière générale, il a été observé que pour des maillages avec moins de 16 cellules dans le diamètre de la goutte, les estimations d'évaporation comportent des erreurs de plus de 50%, ce qui n'est pas acceptable pour des calculs haute-fidélité.
Il est alors important de continuer à travailler sur la compréhension et le développement

de méthodes capables d'obtenir des résultats précis avec un nombre de mailles raisonnable. Cela passe par un travail plus important sur les problèmes de conservation dans le cas des méthodes Level-Set ou dans l'amélioration de la précision des transports scalaires pour les méthodes VOF.

Bibliography

[1] T. Abadie, J. Aubin, and D. Legendre. On the combined effects of surface tension force calculation and interface advection on spurious currents within Volume of Fluid and Level Set frameworks. *Journal of Computational Physics*, 297:611–636, 2015. p. 119, 124, 126, 129, 131, 134

[2] M. O. Abu-Al-Saud, S. Popinet, and H. A. Tchelepi. A conservative and well-balanced surface tension model. *Journal of Computational Physics*, 371:896–913, 2018. p. 106

[3] A. Albadawi, D. B. Donoghue, A. J. Robinson, D. B. Murray, and Y. M. Delauré. Influence of surface tension implementation in Volume of Fluid and coupled Volume of Fluid with Level Set methods for bubble growth and detachment. *International Journal of Multiphase Flow*, 53:11–28, 2013. p. 68

[4] R. Alis. *Simulation numérique directe de gouttes et de groupes de gouttes qui s' évaporent dans un écoulement laminaire ou turbulent*. PhD thesis, Toulouse, ISAE, 2018. p. 5, 6, 184, 212, 237, 254

[5] A. Alke and D. Bothe. 3d numerical modeling of soluble surfactant at fluidic interfaces based on the volume-of-fluid method. *FDMP*, 1:1–29, 2009. p. 182

[6] C. Antoine. Thermodynamic vapor pressures: New relation between the pressures and the temperatures (thermodynamique, tensions des vapeurs: Novelle relation entre les tensions et les temperatures). *CR Hebd. Seances Acad. Sci*, 107(681):836, 1888. p. 22

[7] L. Anumolu and M. F. Trujillo. Gradient augmented level set method for phase change simulations. *Journal of Computational Physics*, 353:377–406, 2018. p. 152, 160, 161, 164, 165, 167, 169, 170, 171, 187, 189, 191, 192, 207, 224, 229

[8] N. Ashgriz and J. Y. Poo. Coalescence and separation in binary collisions of liquid drops. *Journal of Fluid Mechanics*, 221:183–204, 1990. p. 140, 141

[9] T. D. Aslam. A partial differential equation approach to multidimensional extrapolation. *Journal of Computational Physics*, 193(1):349–355, 2004. p. 259, 262

[10] A. Asuri Mukundan, T. Ménard, J. C. Brändle de Motta, and A. Berlemont. A 3D Moment of Fluid method for simulating complex turbulent multiphase flows. *Computers and Fluids*, 198:104364, 2020. p. 142, 143

[11] Y. Atmani, F. Pecquery, M. Cailler, V. Moureau, G. Lartigue, R. Mercier, R. Janodet, G. Sahut, and G. Balarac. Consistent scalar transport with front capturing methods: application to two-phase heat transfer. In *ICLASS 2021, 15th Triennial International Conference on Liquid Atomization and Spray Systems*, 2021. p. 36

[12] E. Aulisa. Tecniche di ricostruzione e di convezione nello spazio tridimensionale del linterfaccia di separazione tra sistemi bifase composti da uidi immiscibili e incomprimibili. *Master thesis, University of Bologna*, 2001. p. 55

[13] E. Aulisa, S. Manservisi, R. Scardovelli, and S. Zaleski. A geometrical area-preserving volume-of-fluid advection method. *Journal of Computational Physics*, 192(1):355–364, 2003. p. 51

[14] S. Balay, W. D. Gropp, L. C. McInnes, and B. F. Smith. Efficient management of parallelism in object oriented numerical software libraries. In E. Arge, A. M. Bruaset, and H. P. Langtangen, editors, *Modern Software Tools in Scientific Computing*, pages 163–202. Birkhäuser Press, 1997. p. 123

[15] R. Banerjee. A numerical study of combined heat and mass transfer in an inclined channel using the vof multiphase model. *Numerical Heat Transfer, Part A: Applications*, 52(2):163–183, 2007. p. 177

[16] Y. Bazilevs, M. Hsu, J. Kiendl, R. Wüchner, and K. Bletzinger. 3D Simulation of Wind Turbine Rotors at Full Scale. Part II: Fluid-Structure Interaction Modeling with Composite Blades. *International Journal for Numerical Methods in Fluids*, 65(October 2010):236–253, 2011. p. 119

[17] S. Bnà, A. Cervone, V. Le Chenadec, S. Manservisi, and R. Scardovelli. Review of split and unsplit geometric advection algorithms. In *AIP Conference Proceedings*, volume 1558, pages 875–878. AIP, 2013. p. 53

[18] P. Bochev and M. Shashkov. Constrained interpolation (remap) of divergence-free fields. *Computer Methods in Applied Mechanics and Engineering*, 194(2-5 SPEC. ISS.):511–530, 2005. p. 97

[19] G. Bornia, A. Cervone, S. Manservisi, R. Scardovelli, and S. Zaleski. On the properties and limitations of the height function method in two-dimensional cartesian geometry. *Journal of Computational Physics*, 230(4):851–862, 2011. p. 46, 111

[20] D. Bothe and S. Fleckenstein. A Volume-of-Fluid-based method for mass transfer processes at fluid particles. *Chemical Engineering Science*, 101:283–302, 2013. p. 102

[21] Z. Bouali, B. Duret, F.-X. Demoulin, and A. Mura. Dns analysis of small-scale turbulence-scalar interactions in evaporating two-phase flows. *International Journal of Multiphase Flow*, 85:326–335, 2016. p. 170

[22] J. U. Brackbill, D. B. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of computational physics*, 100(2):335–354, 1992. p. 106, 108

[23] D. L. Brown, R. Cortez, and M. L. Minion. Accurate Projection Methods for the Incompressible Navier-Stokes Equations. *Journal of Computational Physics*, 168(2):464–499, 2001. p. 91

[24] L. Bureš and Y. Sato. Direct numerical simulation of evaporation and condensation with the geometric VOF method and a sharp-interface phase-change model. *International Journal of Heat and Mass Transfer*, 173, 2021. p. 22, 164, 165, 183, 189

[25] M. Bussmann, D. B. Kothe, and J. M. Sicilian. Modeling high density ratio incompressible interfacial flows. *American Society of Mechanical Engineers, Fluids Engineering Division (Publication) FED*, 257(1 B):707–713, 2002. p. 100, 138

[26] J. W. Cahn and J. E. Hilliard. Free energy of a nonuniform system. i. interfacial free energy. *The Journal of chemical physics*, 28(2):258–267, 1958. p. 3

[27] Z. Cao, J. Zhou, A. Liu, D. Sun, B. Yu, and J. Wei. A three dimensional coupled VOF and Level set (VOSET) method with and without phase change on general curvilinear grids. *Chemical Engineering Science*, 223, 2020. p. 260, 267

[28] A. Cervone, S. Manservisi, and R. Scardovelli. An optimal constrained approach for divergence-free velocity interpolation and multilevel vof method. *Computers & Fluids*, 47(1):101–114, 2011. p. 48

[29] M. Chai, K. Luo, C. Shao, H. Wang, and J. Fan. A coupled vaporization model based on temperature/species gradients for detailed numerical simulations using conservative level set method. *International Journal of Heat and Mass Transfer*, 2018. p. 5, 152, 153, 164, 169, 171, 193, 241, 242

[30] M. Chai, K. Luo, C. Shao, H. Wang, and J. Fan. A finite difference discretization method for heat and mass transfer with Robin boundary conditions on irregular domains. *Journal of Computational Physics*, 400:108890, 2020. p. 194, 208, 209, 212, 264

[31] M. Chai, K. Luo, H. Wang, S. Zheng, and J. Fan. Imposing mixed Dirichlet-Neumann-Robin boundary conditions on irregular domains in a level set/ghost fluid based finite difference framework. *Computers and Fluids*, 214:104772, 2021. p. 209

[32] C. Chauveau, M. Birouk, and I. Gökalp. An analysis of the d2-law departure during droplet evaporation in microgravity. *International Journal of Multiphase Flow*, 37(3):252–259, 2011. p. 2

[33] L. Chen, D. Nguyen, S. Banerjee, and F. Gibou. A level set based sharp interface method for the multiphase incompressible Navier-Stokes equations with phase change. *Journal of Computational Physics*, 222(2):536–555, 2007. p. 101, 152, 165, 169, 171

[34] S. Chen, B. Merriman, S. Osher, and P. Smereka. A Simple Level Set Method for Solving Stefan Problems. *Journal of Computational Physics*, 135(1):8–29, 1997. p. 207

[35] X. Chen and X. Zhang. An improved 2d mof method by using high order derivatives. *Journal of Computational Physics*, 349:176–190, 2017. p. 50

[36] V. L. Chenadec and H. Pitsch. A 3D unsplit Forward/Backward Volume-of-Fluid approach and coupling to the level set method. *Journal of Computational Physics*, 233(1):10–33, 2013. p. 70, 71, 77

[37] A. Chiapolino, R. Saurel, and B. Nkonga. Sharpening diffuse interfaces with compressible fluids on unstructured meshes. *Journal of Computational Physics*, 340:389–417, 2017. p. 26, 44

[38] R. Chiodi and O. Desjardins. A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows. *Journal of Computational Physics*, 343:186–200, 2017. p. 35, 65, 66, 109, 122, 124, 129

[39] A. J. Chorin. On the Convergence of Discrete Approximations to the Navier-Stokes Equations. *Mathematics of Computation*, 23(106):341, 1969. p. 90, 120

[40] A. Clappier. A correction method for use in multidimensional time-splitting advec-

tion algorithms: Application to two- and three-dimensional transport. *Monthly Weather Review*, 126(1):232–242, 1998. p. 102

[41] M. Coquerelle and S. Glockner. A fourth-order accurate curvature computation in a level set framework for two-phase flows subjected to surface tension forces. *Journal of Computational Physics*, 305:838–876, 2016. p. 109

[42] S. J. Cummins, M. M. Francois, and D. B. Kothe. Estimating curvature from volume fractions. *Computers & structures*, 83(6-7):425–434, 2005. p. 109, 112

[43] A. Datta. Porous media approaches to studying simultaneous heat and mass transfer in food processes. ii: Property data and representative results. *Journal of food engineering*, 80(1):96–110, 2007. p. 1

[44] J. M. Delhaye. Jump conditions and entropy sources in two-phase systems. Local instant formulation. *International Journal of Multiphase Flow*, 1(3):395–409, 1974. p. 17

[45] A. D. Demou, N. Scapin, M. Pelanti, and L. Brandt. A pressure-based diffuse interface method for low-Mach multiphase flows with mass transfer. 2021. p. 254

[46] O. Desjardins, G. Blanquart, G. Balarac, and H. Pitsch. High order conservative finite difference scheme for variable density low Mach number turbulent flows. *Journal of Computational Physics*, 227(15):7125–7159, 2008. p. 67

[47] O. Desjardins and V. Moureau. Methods for multiphase flows with high density ratio. *Center for Turbulence Research Proceedings of the Summer Program*, pages 313–322, 2010. p. 99, 102, 139

[48] O. Desjardins, V. Moureau, and H. Pitsch. An accurate conservative level set/ghost fluid method for simulating turbulent atomization. *Journal of Computational Physics*, 227(18):8395–8416, 2008. p. 26, 36, 64, 67, 102

[49] O. Desjardins and H. Pitsch. A spectrally refined interface approach for simulating multiphase flows. *Journal of Computational Physics*, 228(5):1658–1677, 2009. p. 62, 136

[50] S. Diwakar, S. K. Das, and T. Sundararajan. A quadratic spline based interface (quasi) reconstruction algorithm for accurate tracking of two-phase flows. *Journal of Computational Physics*, 228(24):9107–9130, 2009. p. 50

[51] M. Dodd. *Direct numerical simulation of droplet-laden isotropic turbulence*. PhD thesis, 2017. p. 159, 190

[52] M. S. Dodd, D. Mohaddes, A. Ferrante, and M. Ihme. Analysis of droplet evaporation in isotropic turbulence through droplet-resolved DNS. *International Journal of Heat and Mass Transfer*, 172:121157, 2021. p. 5, 22, 154, 168, 169, 254

[53] D. A. Drew. Mathematical modeling of two-phase flow. *Annual review of fluid mechanics*, 15(1):261–291, 1983. p. 28

[54] A. Du Chéné, C. Min, and F. Gibou. Second-order accurate computation of curvatures in a level set framework using novel high-order reinitialization schemes. *Journal of Scientific Computing*, 35(2-3):114–131, 2008. p. 58

[55] V. Dyadechko and M. Shashkov. Moment-of-fluid interface reconstruction. *Los Alamos Report LA-UR-05-7571*, 2005. p. 49

[56] B. Engquist, A. K. Tornberg, and R. Tsai. Discretization of Dirac delta functions in level set methods. *Journal of Computational Physics*, 207(1):28–51, 2005. p. 107

[57] Å. Ervik, K. Y. Lervåg, and S. T. Munkejord. A robust method for calculating interface curvature and normal vectors using an extracted local level set. *Journal of Computational Physics*, 257:259–277, 2014. p. 110

[58] A. Esmaeeli and G. Tryggvason. Computations of film boiling. Part I: Numerical method. *International Journal of Heat and Mass Transfer*, 47(25):5451–5461, 2004. p. 154, 159, 168

[59] R. P. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152(2):457–492, 1999. p. 255

[60] P. A. Ferdowsi and M. Bussmann. Second-order accurate normals from height functions. *Journal of Computational Physics*, 227(22):9293–9302, 2008. p. 111

[61] G. Finotello, J. T. Padding, N. G. Deen, A. Jongsma, F. Innings, and J. A. Kuipers. Effect of viscosity on droplet-droplet collisional interaction. *Physics of Fluids*, 29(6), 2017. p. 104, 140

[62] M. M. Francois, S. J. Cummins, E. D. Dendy, D. B. Kothe, J. M. Sicilian, and M. W. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213(1):141–173, 2006. p. 100, 107

[63] D. Fuster, G. Agbaglah, C. Josserand, S. Popinet, and S. Zaleski. Numerical simulation of droplets, bubbles and waves: state of the art. *Fluid dynamics research*, 41(6):065001, 2009. p. 119

[64] D. Fuster, T. Arrufat, M. Crialesi-Esposito, Y. Ling, L. Malan, S. Pal, R. Scardovelli, G. Tryggvason, and S. Zaleski. A momentum-conserving, consistent, Volume-of-Fluid method for incompressible flow on staggered grids. *arXiv*, 2018. p. 94, 143

[65] D. Gerlach, G. Tomar, G. Biswas, and F. Durst. Comparison of volume-of-fluid methods for surface tension-dominant two-phase flows. *International Journal of Heat and Mass Transfer*, 49(3-4):740–754, 2006. p. 119, 136

[66] S. Ghods and M. Herrmann. A consistent rescaled momentum transport method for simulating large density ratio incompressible multiphase flows using level set methods. *Physica Scripta*, 88(T155), 2013. p. 100, 136

[67] F. Gibou and R. Fedkiw. *A fourth order accurate discretization for the Laplace and heat equations on arbitrary domains, with applications to the Stefan problem*, volume 202. Elsevier, 2005. p. 57, 207, 224, 226

[68] F. Gibou, R. P. Fedkiw, L. T. Cheng, and M. Kang. A second-order-accurate symmetric discretization of the Poisson equation on irregular domains. *Journal of Computational Physics*, 176(1):205–227, 2002. p. 169, 207, 212, 215

[69] J. P. Gois, A. Nakano, L. G. Nonato, and G. C. Buscaglia. Front tracking with moving-least-squares surfaces. *Journal of Computational Physics*, 227(22):9643–9669, 2008. p. 114

[70] R. Goldman. Curvature formulas for implicit curves and surfaces. *Computer Aided Geometric Design*, 22(7 SPEC. ISS.):632–658, 2005. p. 109

[71] B. E. Griffith. On the volume conservation of the immersed boundary method. *Communications in Computational Physics*, 12(2):401–432, 2012. p. 93

[72] G. Guédon. Two-Phase Heat and Mass Transfer Modeling: Flexible Numerical Meth-

ods for Energy Engineering Analyses. (August):132, 2013. p. 190

[73] D. Gueyffier, J. Li, A. Nadim, R. Scardovelli, and S. Zaleski. Volume-of-fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational physics*, 152(2):423–456, 1999. p. 52, 136

[74] D. Z. Guo, D. L. Sun, Z. Y. Li, and W. Q. Tao. Phase change heat transfer simulation for boiling bubbles arising from a vapor film by the VOSET method. *Numerical Heat Transfer; Part A: Applications*, 59(11):857–881, 2011. p. 156, 243

[75] Z. Guo, D. Fletcher, and B. Haynes. A review of computational modelling of flow boiling in microchannels. *The Journal of Computational Multiphase Flows*, 6(2):79–110, 2014. p. 178

[76] S. Hardt and F. Wondra. Evaporation model for interfacial flows based on a continuum-field representation of the source terms. *Journal of Computational Physics*, 227(11):5871–5895, 2008. p. 155, 224, 229

[77] F. H. Harlow and J. E. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *The physics of fluids*, 8(12):2182–2189, 1965. p. 45, 91

[78] D. Hartmann, M. Meinke, and W. Schröder. The constrained reinitialization equation for level set methods. *Journal of Computational Physics*, 229(5):1514–1535, 2010. p. 59

[79] J. Hernández, J. López, P. Gómez, C. Zanzi, and F. Faura. A new volume of fluid method in three dimensionsâ€”part i: Multidimensional advection method with face-matched flux polyhedra. *International Journal for Numerical Methods in Fluids*, 58(8):897–921, 2008. p. 112

[80] M. Herrmann. A domain decomposition parallelization of the Fast Marching Method. *Annual Research Briefs, Center for Turbulence Research, NASA*, pages 213–225, 2003. p. 61, 262

[81] M. Herrmann. A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids. *Journal of Computational Physics*, 227(4):2674–2706, 2008. p. 100, 104, 121, 146

[82] M. Herrmann, G. Blanquart, and V. Raman. Flux corrected finite volume scheme for preserving scalar boundedness in reacting large-eddy simulations. *AIAA journal*, 44(12):2879–2886, 2006. p. 65, 68, 76

[83] C. W. Hirt and B. D. Nichols. Volume of fluid (vof) method for the dynamics of free boundaries. *Journal of computational physics*, 39(1):201–225, 1981. p. 4, 45

[84] S. Ii, K. Sugiyama, S. Takeuchi, S. Takagi, Y. Matsumoto, and F. Xiao. An interface capturing method with a continuous function: The THINC method with multi-dimensional reconstruction. *Journal of Computational Physics*, 231(5):2328–2358, 2012. p. 26, 44

[85] M. Irfan and M. Muradoglu. A front tracking method for direct numerical simulation of evaporation process in a multiphase system. *Journal of Computational Physics*, 337:132–153, 2017. p. 5, 153, 154, 168, 169, 190

[86] M. Irfan and M. Muradoglu. A front tracking method for particle-resolved simulation of evaporation and combustion of a fuel droplet. *Computers and Fluids*, 174:283–299, 2018. p. 5

[87] M. Ishii and T. Hibiki. *Thermo-fluid dynamics of two-phase flow*. Springer Science & Business Media, 2010. p. 11

[88] D. Jacqmin. Calculation of two-phase navier–stokes flows using phase-field modeling. *Journal of computational physics*, 155(1):96–127, 1999. p. 35

[89] R. Janodet, C. Guillamón, V. Moureau, R. Mercier, G. Lartigue, P. Benard, T. Ménard, A. Berlemont, R. Janodet, C. Guillamón, V. Moureau, R. Mercier, G. Lartigue, V. Moureau, and R. Mercier. A massively parallel accurate conservative level set algorithm for simulating turbulent atomization on adaptive unstructured grids To cite this version : HAL Id : hal-03024186 A massively parallel accurate conservative level set algorithm for simulating t. 2020. p. 66, 67, 76

[90] R. Janodet, G. Vaudor, G. Lartigue, P. Bénard, V. Moureau, and R. Mercier. An unstructured conservative level-set algorithm coupled with dynamic mesh adaptation for the computation of liquid-gas flows. *preprint on HAL*, 2019. p. 77, 136, 141, 146

[91] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, and Y. Wang. A coupled level set-moment of fluid method for incompressible two-phase flows. *Journal of Scientific Computing*, 54(2-3):454–491, 2013. p. 69

[92] G. S. Jiang and D. Peng. Weighted ENO schemes for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, 21(6):2126–2143, 2000. p. 57, 61, 75

[93] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted eno schemes. *Journal of computational physics*, 126(1):202–228, 1996. p. 62, 75, 94

[94] W. Jones and K. Menzies. Analysis of the cell-centred finite volume method for the diffusion equation. *Journal of Computational Physics*, 165(1):45–68, 2000. p. 208

[95] D. Juric and G. Tryggvason. Computations of boiling flows. *International journal of multiphase flow*, 24(3):387–410, 1998. p. 5

[96] D. Kah, V. Mittal, and P. Shashank. Les of spray and combustion in an internal combustion engine. *Center for Turbulence Research, Annual Research Briefs*, 2012. p. 2

[97] M. Kang, R. P. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15(3):323–360, 2000. p. 104, 108

[98] I. Kikic and A. Vetere. Evaluation of several literature equations to predict the vaporization enthalpies at the normal boiling point. *Fluid phase equilibria*, 309(2):151–154, 2011. p. 22

[99] J. Kim. Spray cooling heat transfer: The state of the art. *International Journal of Heat and Fluid Flow*, 28(4):753–767, 2007. p. 1

[100] J. Kim and P. Moin. Application of a fractional-step method to incompressible navier-stokes equations. *Journal of computational physics*, 59(2):308–323, 1985. p. 91

[101] M. Kuhn and O. Desjardins. An all-Mach, low-dissipation strategy for simulating multiphase flows. *Journal of Computational Physics*, page 110602, 2021. p. 5, 254

[102] C. Kunkelmann. *Numerical modeling and investigation of boiling phenomena*. PhD thesis, 2011. p. 182, 190, 260, 263

[103] C. Kunkelmann and P. Stephan. Modification and Extension of a Standard Volume of Fluid Solver For Simulating Boiling Heat Transfer. *V European Conference on*

Computational Fluid Dynamics, (June):14–17, 2010. p. 168, 169

[104] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with surfer. *Journal of computational physics*, 113(1):134–147, 1994. p. 102

[105] B. Lalanne, L. R. Villegas, S. Tanguy, and F. Risso. On the computation of viscous terms for incompressible two-phase flows with Level Set/Ghost Fluid Method. *Journal of Computational Physics*, 301:289–307, 2015. p. 102, 104, 105, 106

[106] V. Le Chenadec and H. Pitsch. A monotonicity preserving conservative sharp interface flow solver for high density ratio two-phase flows. *Journal of Computational Physics*, 2013. p. 103

[107] M. S. Lee, A. Riaz, and V. Aute. Direct numerical simulation of incompressible multiphase flow with phase change. *Journal of Computational Physics*, 344:381–418, 2017. p. 35, 155, 157, 158, 159, 169, 171

[108] A. Lemoine, S. Glockner, and J. Breil. Moment-of-fluid analytic reconstruction on 2d cartesian grids. *Journal of Computational Physics*, 328:131–139, 2017. p. 50

[109] B. P. Leonard. A stable and accurate convective modelling procedure based on quadratic upstream interpolation. *Computer methods in applied mechanics and engineering*, 19(1):59–98, 1979. p. 94

[110] J. Leparoux, R. Mercier, V. Moureau, and H. Musaefendic. Primary atomization simulation applied to a jet in crossflow aeronautical injector with dynamic mesh adaptation. In *14th Triennal International Conference on Liquid Atomization and Spray Systems*, page 5, 2018. p. 4

[111] R. J. LeVeque. High-resolution conservative algorithms for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33(2):627–665, 1996. p. 81, 83

[112] R. J. LeVeque et al. *Finite volume methods for hyperbolic problems*, volume 31. Cambridge university press, 2002. p. 67

[113] J. Li. Calcul d'interface affine par morceaux. *Comptes rendus de l'Académie des sciences. Série II, Mécanique, physique, chimie, astronomie*, 320(8):391–396, 1995. p. 52

[114] P. Liovic, M. Francois, M. Rudman, and R. Manasseh. Efficient simulation of surface tension-dominated flows through enhanced interface geometry interrogation. *Journal of Computational Physics*, 229(19):7520–7544, 2010. p. 112

[115] P. Liovic, M. Rudman, J.-L. Liow, D. Lakehal, and D. Kothe. A 3d unsplit-advection volume tracking algorithm with planarity-preserving interface reconstruction. *Computers & fluids*, 35(10):1011–1032, 2006. p. 48

[116] X. Liu, K. Morita, and S. Zhang. Direct numerical simulation of incompressible multiphase flow with vaporization using moving particle semi-implicit method. *Journal of Computational Physics*, 425:109911, 2021. p. 154

[117] X. D. Liu, R. P. Fedkiw, and M. Kang. A Boundary Condition Capturing Method for Poisson's Equation on Irregular Domains. *Journal of Computational Physics*, 160(1):151–178, 2000. p. 255

[118] J. López, J. Hernández, P. Gómez, and F. Faura. A volume of fluid method based on multidimensional advection and spline interface reconstruction. *Journal of Computational Physics*, 195(2):718–742, 2004. p. 51

[119] J. López, C. Zanzi, P. Gómez, R. Zamora, F. Faura, and J. Hernández. An improved height function technique for computing interface curvature from volume fractions. *Computer Methods in Applied Mechanics and Engineering*, 198(33-36):2555–2564, 2009. p. 112

[120] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3d surface construction algorithm. *ACM siggraph computer graphics*, 21(4):163–169, 1987. p. 183

[121] X. Y. Luo, M. J. Ni, A. Ying, and M. A. Abdou. Numerical modeling for multiphase incompressible flow with phase change. *Numerical Heat Transfer, Part B: Fundamentals*, 48(5):425–444, 2005. p. 154

[122] C. Ma and D. Bothe. Numerical modeling of thermocapillary two-phase flows with evaporation using a two-scalar approach for heat transfer. *Journal of Computational Physics*, 233:552–573, 2013. p. 31, 151, 152, 156, 160, 168, 169, 171, 172, 198, 201, 243

[123] L. Malan, A. Malan, S. Zaleski, and P. Rousseau. A geometric vof method for interface resolved phase change and conservative thermal energy advection. *Journal of Computational Physics*, 426:109920, 2021. p. 152, 163, 164, 166, 168, 169, 171, 190, 263

[124] L. C. Malan, R. Scardovelli, and S. Zaleski. Using Extrapolation Techniques in VOF Methodology to Model Expanding Bubbles. *Procedia IUTAM*, 15:228–235, 2015. p. 161

[125] E. Marchandise, P. Geuzaine, N. Chevaugeon, and J. F. Remacle. A stabilized finite element method using a discontinuous level set approach for the computation of bubble dynamics. *Journal of Computational Physics*, 225(1):949–974, 2007. p. 109

[126] L. G. Martinez, B. Duret, J. Reveillon, and F. X. Demoulin. International Journal of Multiphase Flow A new DNS formalism dedicated to turbulent two-phase flows with phase change. *International Journal of Multiphase Flow*, 143(July):103762, 2021. p. 5, 6, 170, 171, 254

[127] J. O. McCaslin, É. Courtine, and O. Desjardins. A fast marching approach to multidimensional extrapolation. *Journal of Computational Physics*, 274:393–412, 2014. p. 61, 262, 263, 267

[128] J. O. McCaslin and O. Desjardins. A localized re-initialization equation for the conservative level set method. *Journal of Computational Physics*, 262:408–426, 2014. p. 65

[129] P. McCorquodale, P. Colella, and H. Johansen. A Cartesian grid embedded boundary method for the heat equation on irregular domains. *Journal of Computational Physics*, 173(2):620–635, 2001. p. 203, 204, 206

[130] D. Mercier. *Large eddy simulation of coupled dispersed phase flows: a statistically-consistent formalism*. PhD thesis, Université Paris-Saclay, 2020. p. 1

[131] M. Meyer, A. Devesa, S. Hickel, X. Y. Hu, and N. A. Adams. A conservative immersed interface method for Large-Eddy Simulation of incompressible flows. *Journal of Computational Physics*, 229(18):6300–6317, 2010. p. 206

[132] M. Mikami, H. Oyagi, N. Kojima, M. Kikuchi, Y. Wakashima, and S. Yoda. Microgravity experiments on flame spread along fuel-droplet arrays using a new

droplet-generation technique. *Combustion and flame*, 141(3):241–252, 2005. p. 2

[133] G. Miller and P. Colella. A conservative three-dimensional eulerian method for coupled solid–fluid shock capturing. *Journal of Computational Physics*, 183(1):26–82, 2002. p. 48, 84

[134] C. Min and F. Gibou. A second order accurate level set method on non-graded adaptive cartesian grids. *Journal of Computational Physics*, 225(1):300–321, 2007. p. 62, 183

[135] S. Mirjalili, S. S. Jain, and M. Dodd. Interface-capturing methods for two-phase flows: An overview and recent developments. *Center for Turbulence Research Annual Research Briefs*, pages 117–135, 2017. p. 47

[136] S. Mirjalili and A. Mani. Consistent, energy-conserving momentum transport for simulations of two-phase flows using the phase field equations. *Journal of Computational Physics*, 426:109918, 2021. p. 35

[137] R. Mittal and G. Iaccarino. Immersed boundary methods. *Annual Review of Fluid Mechanics*, 37:239–261, 2005. p. 201

[138] M. Motamed, C. B. Macdonald, and S. J. Ruuth. On the linear stability of the fifth-order weno discretization. *Journal of Scientific Computing*, 47(2):127–149, 2011. p. 102

[139] A. A. Mukundan, T. Ménard, A. Berlemont, J. César, B. D. Motta, and C. Umrcoria. Coupled Level set moment of fluid method for simulating multiphase flows. (September):2–4, 2019. p. 69, 71

[140] A. A. Mukundan, T. Menard, A. Berlemont, and J. C. B. de Motta. Numerical study of interface reconstruction method in under-resolved regions of the flow for liquid jet primary breakup. 2018. p. 50

[141] A. A. Mukundan, T. Ménard, A. Berlemont, and J. C. B. de Motta. A comparative study of dns of airblast atomization using clsmof and clsvof methods. In *29th Annual Conference on Liquid Atomization and Spray Systems (ILASS19)*, 2019. p. 69

[142] N. Nangia, B. E. Griffith, N. A. Patankar, and A. P. S. Bhalla. A robust incompressible Navier-Stokes solver for high density ratio multiphase flows. *Journal of Computational Physics*, 390:548–594, 2019. p. 104

[143] J. C. Nave, R. R. Rosales, and B. Seibold. A gradient-augmented level set method with an optimally local, coherent advection scheme. *Journal of Computational Physics*, 229(10):3802–3827, 2010. p. 62

[144] Y. T. Ng, C. Min, and F. Gibou. An efficient fluid-solid coupling algorithm for single-phase flows. *Journal of Computational Physics*, 228(23):8807–8829, 2009. p. 206

[145] D. Q. Nguyen, R. P. Fedkiw, and M. Kang. A boundary condition capturing method for incompressible flame discontinuities. *Journal of Computational Physics*, 172(1):71–98, 2001. p. 156, 158, 160

[146] B. Nichols, C. Hirt, and R. Hotchkiss. Sola-vof: A solution algorithm for transient fluid flow with multiple free boundaries. Technical report, Los Alamos Scientific Lab., 1980. p. 46

[147] B. M. Ningegowda and B. Premachandran. A Coupled Level Set and Volume of Fluid method with multi-directional advection algorithms for two-phase flows with and

without phase change. *International Journal of Heat and Mass Transfer*, 79:532–550, 2014. p. 254

[148] W. F. Noh and P. Woodward. Slic (simple line interface calculation). In *Proceedings of the fifth international conference on numerical methods in fluid dynamics June 28–July 2, 1976 Twente University, Enschede*, pages 330–340. Springer, 1976. p. 26, 45

[149] R. R. Nourgaliev and T. G. Theofanous. High-fidelity interface tracking in compressible flows: Unlimited anchored adaptive level set. *Journal of Computational Physics*, 224(2):836–866, 2007. p. 62, 119

[150] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *Journal of computational physics*, 210(1):225–246, 2005. p. 26, 35, 62, 67

[151] E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *Journal of Computational Physics*, 2007. p. 64, 67

[152] S. Osher and R. Fedkiw. *Level set methods and dynamic implicit surfaces*, volume 153. Springer Science & Business Media, 2006. p. 75

[153] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: algorithms based on hamilton-jacobi formulations. *Journal of computational physics*, 79(1):12–49, 1988. p. 4, 26, 56

[154] M. Owkes, E. Cauble, J. Senecal, and R. A. Currie. Importance of curvature evaluation scale for predictive simulations of dynamic gas â€“ liquid interfaces. *Journal of Computational Physics*, 365:37–55, 2018. p. 115

[155] M. Owkes and O. Desjardins. A discontinuous Galerkin conservative level set scheme for interface capturing in multiphase flows. *Journal of Computational Physics*, 2013. p. 67, 112

[156] M. Owkes and O. Desjardins. A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (vof) method. *Journal of Computational Physics*, 270:587–612, 2014. p. 48, 51

[157] M. Owkes and O. Desjardins. A mesh-decoupled height function method for computing interface curvature. *Journal of Computational Physics*, 281:285–300, 2015. p. 112, 114, 115

[158] M. Owkes and O. Desjardins. A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows. *Journal of Computational Physics*, 2017. p. 97

[159] J. Palmore and O. Desjardins. A volume of fluid framework for interface-resolved simulations of vaporizing liquid-gas flows. *Journal of Computational Physics*, 399:108954, 2019. p. 5, 99, 121, 151, 152, 154, 155, 160, 161, 162, 164, 165, 171, 172, 190, 196, 198, 205, 206, 213, 214, 220, 222, 223, 224, 229, 230, 231, 235, 239, 254

[160] J. Papac, F. Gibou, and C. Ratsch. Efficient symmetric discretization for the Poisson, heat and Stefan-type problems with Robin boundary conditions. *Journal of Computational Physics*, 229(3):875–889, 2010. p. 171, 205, 206, 212

[161] B. Parker and D. Youngs. *Two and three dimensional Eulerian simulation of fluid flow with material interfaces*. Atomic Weapons Establishment, 1992. p. 46

[162] H. V. Patel, S. Das, J. A. Kuipers, J. T. Padding, and E. A. Peters. A coupled

Volume of Fluid and Immersed Boundary Method for simulating 3D multiphase flows with contact line dynamics in complex geometries. *Chemical Engineering Science*, 166:28–41, 2017. p. 206

[163] A. Pathak and M. Raessi. An implicit, sharp numerical treatment of viscous terms at arbitrarily shaped liquid-gas interfaces in evaporative flows. *Journal of Computational Physics*, 418:109625, 2020. p. 104, 158

[164] M. Pelletier. *Diffuse interface models and adapted numerical schemes for the simulation of subcritical to supercritical flows*. PhD thesis, Université Paris-Saclay (ComUE), 2019. p. 3

[165] R. B. Pember, J. B. Bell, P. Colella, W. Y. Curtchfield, and M. L. Welcome. An adaptive cartesian grid method for unsteady compressible flow in irregular regions. *Journal of computational Physics*, 120(2):278–304, 1995. p. 203

[166] C. S. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25(3):220–252, 1977. p. 202

[167] J. E. Pilliod. *An analysis of piecewise linear interface reconstruction algorithms for volume-of-fluid methods*. U. of Calif., Davis., 1992. p. 48

[168] J. E. Pilliod Jr and E. G. Puckett. Second-order accurate volume-of-fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199(2):465–502, 2004. p. 26, 45

[169] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228(16):5838–5866, 2009. p. 112, 114, 115, 122, 125, 129, 133, 134, 136, 146

[170] S. Popinet. Numerical models of surface tension. *Annual Review of Fluid Mechanics*, 50:49–75, 2018. p. 107

[171] S. Popinet and collaborators. Basilisk. http://basilisk.fr/src/fractions.h. p. 180

[172] S. Popinet and S. Zaleski. A front-tracking algorithm for accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, 30(6):775–793, 1999. p. 4, 136

[173] A. Prosperetti. Motion of two superposed viscous fluids. *The Physics of Fluids*, 24(7):1217–1223, 1981. p. 137

[174] E. G. Puckett. A volume-of-fluid interface tracking algorithm with applications to computing shock wave refraction. In *Proceedings of the Fourth International Symposium on Computational Fluid Dynamics*, pages 933–938, 1991. p. 48

[175] J. W. Purvis and J. E. Burkhalter. Prediction of critical mach number for store configurations. *AIAA Journal*, 17(11):1170–1177, 1979. p. 202

[176] M. Raessi and H. Pitsch. Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method. *Computers and Fluids*, 63:70–81, 2012. p. 93

[177] R. C. Reid, J. M. Prausnitz, and B. E. Poling. The properties of gases and liquids. 1987. p. 22

[178] Y. Renardy and M. Renardy. Prost: a parabolic reconstruction of surface tension for the volume-of-fluid method. *Journal of computational physics*, 183(2):400–421, 2002. p. 50, 108, 116

[179] J. Reutzsch, C. Kieffer-Roth, and B. Weigand. A consistent method for direct nu-

merical simulation of droplet evaporation. *Journal of Computational Physics*, 413:109455, 2020. p. 5, 22, 156, 243, 244

[180] J. Rickett and S. Fomel. A second-order fast marching eikonal solver a a Published in SEP Report. *SEP Report*, 100(February 2001):287–292, 1999. p. 59

[181] W. J. Rider and D. B. Kothe. Reconstructing volume tracking. *Journal of computational physics*, 141(2):112–152, 1998. p. 4, 26, 45, 50, 51

[182] M. Rudman. A volume-tracking method for incompressible multifluid flows with large density variations. *International Journal for numerical methods in fluids*, 28(2):357–378, 1998. p. 95, 96, 101

[183] L. Rueda Villegas, R. Alis, M. Lepilliez, and S. Tanguy. A Ghost Fluid/Level Set Method for boiling flows and liquid evaporation: Application to the Leidenfrost effect. *Journal of Computational Physics*, 316:789–813, 2016. p. 5, 61, 75, 152, 153, 158, 160, 162, 164, 169, 170, 171, 223, 237, 238, 239, 246, 254

[184] G. Russo and P. Smereka. A Remark on Computing Distance Functions. *Journal of Computational Physics*, 163(1):51–67, 2000. p. 57, 58

[185] V. Sabelnikov, A. Y. Ovsyannikov, and M. Gorokhovski. Modified level set equation and its numerical assessment. *Journal of Computational Physics*, 278:1–30, 2014. p. 57

[186] G. Sahut. *Simulation numérique de l'ébullition sur maillages non structurés*. PhD thesis, Grenoble Alpes, 2019. p. 65

[187] G. Sahut, G. Ghigliotti, G. Balarac, M. Bernard, V. Moureau, and P. Marty. Numerical simulation of boiling on unstructured grids. *Journal of Computational Physics*, 432:110161, 2021. p. 152, 160, 161, 165, 264

[188] S. Saincher and J. Banerjee. A redistribution-based volume-preserving plic-vof technique. *Numerical Heat Transfer, Part B: Fundamentals*, 67(4):338–362, 2015. p. 56

[189] Y. Sato and B. Ničeno. A conservative local interface sharpening scheme for the constrained interpolation profile method. *International Journal for Numerical Methods in Fluids*, 70(4):441–467, 2012. p. 64

[190] Y. Sato and B. Ničeno. A sharp-interface phase change model for a mass-conservative interface tracking method. *Journal of Computational Physics*, 249:127–161, 2013. p. 152, 154, 155, 166, 169, 170, 171, 189

[191] R. Saurel and R. Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21(3):1115–1145, 1999. p. 3, 26

[192] N. Scapin, P. Costa, and L. Brandt. A volume-of-fluid method for interface-resolved simulations of phase-changing two-fluid flows. *Journal of Computational Physics*, 407:109251, 2019. p. 5, 152, 153, 154, 160, 164, 166, 167, 168, 169, 193, 260, 267

[193] R. Scardovelli and S. Zaleski. Analytical relations connecting linear interfaces and volume fractions in rectangular grids. *Journal of Computational Physics*, 164(1):228–237, 2000. p. 50, 52, 53, 74, 181

[194] R. Scardovelli and S. Zaleski. Interface reconstruction with least-square fit and split eulerian–lagrangian advection. *International Journal for Numerical Methods in Fluids*, 41(3):251–274, 2003. p. 53

[195] J. Schlottke and B. Weigand. Direct numerical simulation of evaporating droplets.

Journal of Computational Physics, 227(10):5215–5237, 2008. p. 5, 22, 152, 153, 156, 166, 168, 169, 189, 243

[196] P. Schwartz, M. Barad, P. Colella, and T. Ligocki. 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, 2006. p. 203, 206

[197] J. A. Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999. p. 59, 262

[198] J. Shaikh, A. Sharma, and R. Bhardwaj. On sharp-interface level-set method for heat and/or mass transfer induced Stefan problem. *International Journal of Heat and Mass Transfer*, 96:458–473, 2016. p. 5, 153, 170, 171, 193, 261

[199] C.-W. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of computational physics*, 77(2):439–471, 1988. p. 75, 76

[200] R. K. Shukla, C. Pantano, and J. B. Freund. An interface capturing method for the simulation of multi-phase compressible flows. *Journal of Computational Physics*, 229(19):7411–7439, 2010. p. 26, 44

[201] W. A. Sirignano. Fuel droplet vaporization and spray combustion theory. *Progress in Energy and Combustion Science*, 9(4):291–322, 1983. p. 1

[202] G. Y. Soh, G. H. Yeoh, and V. Timchenko. An algorithm to calculate interfacial area for multiphase mass transfer through the volume-of-fluid method. *International Journal of Heat and Mass Transfer*, 100:573–581, 2016. p. 178, 179

[203] Z. Solomenko, P. D. Spelt, L. Ó Náraigh, and P. Alix. Mass conservation and reduction of parasitic interfacial waves in level-set methods for the numerical simulation of two-phase flows: A comparative study. *International Journal of Multiphase Flow*, 95:235–256, 2017. p. 61, 119

[204] G. Son and V. K. Dhir. Numerical simulation of film boiling near critical pressures with a level set method. 1998. p. 5, 152, 154

[205] G. Son and N. Hur. A coupled level set and volume-of-fluid method for the buoyancy-driven motion of fluid particles. *Numerical Heat Transfer: Part B: Fundamentals*, 42(6):523–542, 2002. p. 69, 71, 151, 152, 169, 189, 222, 231

[206] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5(3):506–517, 1968. p. 51

[207] G. Strotos, M. Gavaises, A. Theodorakakos, and G. Bergeles. Numerical investigation of the evaporation of two-component droplets. *Fuel*, 90(4):1492–1507, 2011. p. 177

[208] D. L. Sun and W. Q. Tao. A coupled volume-of-fluid and level set (VOSET) method for computing incompressible two-phase flows. *International Journal of Heat and Mass Transfer*, 53(4):645–655, 2010. p. 68

[209] Y. Sun and C. Beckermann. Sharp interface tracking using the phase-field equation. *Journal of Computational Physics*, 220(2):626–653, 2007. p. 35

[210] M. Sussman. A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187(1):110–136, 2003. p. 111

[211] M. Sussman and E. Fatemi. An efficient, interface-preserving level set redistancing algorithm and its application to interfacial incompressible fluid flow. *SIAM Journal on scientific computing*, 20(4):1165–1191, 1999. p. 58

[212] M. Sussman and M. Ohta. A stable and efficient method for treating surface tension in incompressible two-phase flow. *SIAM Journal on Scientific Computing*, 31(4):2447–2471, 2009. p. 112, 122

[213] M. Sussman and E. G. Puckett. A Coupled Level Set and Volume-of-Fluid Method for Computing 3D and Axisymmetric Incompressible Two-Phase Flows. *Journal of Computational Physics*, 162(2):301–337, 2000. p. 68, 70, 71, 102

[214] M. Sussman, P. Smereka, S. Osher, et al. A level set approach for computing solutions to incompressible two-phase flow. 1994. p. 26, 35, 56, 58, 62, 102, 103

[215] M. Sussman, K. M. Smith, M. Y. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *Journal of computational physics*, 221(2):469–505, 2007. p. 101, 103

[216] S. Tanguy. *Développement d'une méthode de suivi d'interface. Applications aux écoulements diphasiques*. PhD thesis, Université de Rouen, 2004. p. 75

[217] S. Tanguy and A. Berlemont. Application of a level set method for simulation of droplet collisions. *International journal of multiphase flow*, 31(9):1015–1035, 2005. p. 142

[218] S. Tanguy, T. Ménard, and A. Berlemont. A Level Set Method for vaporizing two-phase flows. *Journal of Computational Physics*, 2007. p. 5, 69, 70, 71, 77, 101, 152, 153, 156, 158, 160, 161, 162, 165, 170, 171, 193, 242

[219] S. Tanguy, M. Sagan, B. Lalanne, F. Couderc, and C. Colin. Benchmarks and numerical methods for the simulation of boiling flows. *Journal of Computational Physics*, 264:1–22, 2014. p. 152, 154, 155, 160, 161, 165, 169, 170, 171, 248

[220] G. Tomar, G. Biswas, A. Sharma, and A. Agrawal. Numerical simulation of bubble growth in film boiling using a coupled level-set and volume-of-fluid method. *Physics of Fluids*, 17(11):1–13, 2005. p. 254

[221] G. Tóth and P. L. Roe. Divergence- and Curl-preserving prolongation and restriction formulas. *Journal of Computational Physics*, 180(2):736–750, 2002. p. 97

[222] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, and Y. J. Jan. A Front-Tracking Method for the Computations of Multiphase Flow. *Journal of Computational Physics*, 169(2):708–759, 2001. p. 4, 106

[223] S. O. Unverdi and G. Tryggvason. A front-tracking method for viscous, incompressible, multi-fluid flows. *Journal of Computational Physics*, 100(1):25–37, 1992. p. 3

[224] G. Vaudor, T. Ménard, W. Aniszewski, M. Doring, and A. Berlemont. A consistent mass and momentum flux computation method for two phase flows. Application to atomization process. *Computers and Fluids*, 152:204–216, 2017. p. 95, 97, 101, 102

[225] T. Wacławczyk. A consistent solution of the re-initialization equation in the conservative level-set method. *Journal of Computational Physics*, 299:487–525, 2015. p. 65

[226] Y. Wang, S. Simakhina, and M. Sussman. A hybrid level set-volume constraint method for incompressible two-phase flow. *Journal of Computational Physics*, 231(19):6438–6471, 2012. p. 62

[227] S. W. Welch and J. Wilson. A Volume of Fluid Based Method for Fluid Flows with Phase Change. *Journal of Computational Physics*, 160(2):662–682, 2000. p. 5,

154

[228] G. D. Weymouth, D. G. Dommermuth, K. Hendrickson, and D. K.-P. Yue. Advancements in cartesian-grid methods for computational ship hydrodynamics. 2006. p. 55

[229] G. D. Weymouth and D. K.-P. Yue. Conservative volume-of-fluid method for free-surface simulations on cartesian-grids. *Journal of Computational Physics*, 229(8):2853–2865, 2010. p. 55

[230] M. W. Williams. *Numerical methods for tracking interfaces in surface tension in 3-D mold-filling processes*. University of California, Davis, 2000. p. 110

[231] F. Xiao, Y. Honma, and T. Kono. A simple algebraic interface capturing scheme using hyperbolic tangent function. *International Journal for Numerical Methods in Fluids*, 48(9):1023–1040, 2005. p. 26, 44

[232] B. Xie, P. Jin, and F. Xiao. An unstructured-grid numerical model for interfacial multiphase fluids based on multi-moment finite volume formulation and thinc method. *International journal of multiphase flow*, 89:375–398, 2017. p. 44

[233] J. Yang and F. Stern. A highly scalable massively parallel fast marching method for the Eikonal equation. *Journal of Computational Physics*, 332:333–362, 2017. p. 61

[234] S. T. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of computational physics*, 31(3):335–362, 1979. p. 80

[235] B. Zhang, S. Popinet, and Y. Ling. Modeling and detailed numerical simulation of the primary breakup of a gasoline surrogate jet under non-evaporative operating conditions. *International Journal of Multiphase Flow*, 130:103362, 2020. p. 100

[236] J. Zhang and M. J. Ni. Direct numerical simulations of incompressible multiphase magnetohydrodynamics with phase change. *Journal of Computational Physics*, 375:717–746, 2018. p. 152, 154, 155, 156, 160, 166, 168, 169, 190

[237] H. Zinjala and J. Banerjee. A lagrangian-eulerian advection scheme with moment-of-fluid interface reconstruction. *Numerical Heat Transfer, Part B: Fundamentals*, 69(6):563–574, 2016. p. 50

[238] D. Zuzio, A. Orazzo, J. L. Estivalèzes, and I. Lagrange. A new efficient momentum preserving Level-Set/VOF method for high density and momentum ratio incompressible two-phase flows. *Journal of Computational Physics*, 410:109342, 2020. p. 69, 99, 101, 102, 139

université PARIS-SACLAY

ÉCOLE DOCTORALE
Sciences mécaniques et énergétiques, matériaux et géosciences (SMEMAG)

Titre : Simulation numérique d'écoulements diphasiques évaporants par des méthodes de capture d'interface raides

Mots clés : Écoulements diphasiques, Interface raide, Évaporation, Méthodes numérique

Résumé : La simulation des écoulements diphasiques évaporant est un sujet en plein essor avec l'émergence de multiples solveurs et méthodes numériques associés.
C'est un sujet difficile en raison des discontinuités de l'écoulement à l'interface qui nécessitent des méthodes numériques adaptées.
Parmi les méthodes Eulériennes « sharp », plusieurs stratégies ont montré leur capacité à reproduire la physique sans changement de phase, que ce soit en Volume-of-Fluid (VOF) ou en Level-Set (LS).
Avec le changement de phase, les sauts d'interface doivent tenir compte des transferts de masse et de chaleur. Peu de contributions existent, et des études spécifiques sont nécessaires pour atteindre le même niveau de compréhension que pour les écoulements sans évaporation.
L'objectif de ce travail est de comparer les solutions possibles pour simuler de tels écoulements. Toutes les méthodes sont implémentées dans le même solveur et partagent le même cadre unifié Low-Mach pour permettre des comparaisons équitables.

Title : On the numerical simulation of evaporating two-phase flows using sharp interface capturing methods

Keywords : Two-phase flows, Sharp interface, Evaporation, Numerical methods

Abstract : High fidelity simulation of evaporating two-phase flows is a growing subject of interest with the emergence of multiple solvers and various associated numerical methods.
This is a challenging subject because of the strong discontinuities of flow quantities at the interface that require attention in numerical methods design.
For sharp Eulerian methods, several strategies have shown their capability to reproduce the physics without phase change using Volume-of-fluid (VOF) or Level-set (LS).
With phase change, the interface jumps must account for the mass and heat transfers, which requires far more attention numerical methods design. Few contributions can be found, and dedicated studies are necessary to reach the same level of understanding as for non-evaporating flows.
The objective of this work is to compare possible solutions for simulating such flows with sharp Eulerian methods. To this end, they are all implemented in the same cartesian-grid solver and share the same unified Low-Mach framework to allow fair comparisons.