Numerical simulation of primary atomization in aeronautical injectors using a massively parallel adaptive mesh refinement technique

Romain Janodet

Normandie Université

THÈSE

Pour obtenir le diplôme de doctorat

Spécialité Energétique

Préparée à l'INSA de Rouen Normandie

NUMERICAL SIMULATION OF PRIMARY ATOMIZATION IN AERONAUTICAL INJECTORS USING A MASSIVELY PARALLEL ADAPTIVE MESH REFINEMENT TECHNIQUE

présentée et soutenue par

ROMAIN JANODET

Thèse soutenue publiquement le 23 Juin 2022 devant le jury composé de		
S. VINCENT	Professeur, Université Paris-Est Marne-la-Vallée, MSME UMR 8208	Rapporteur
S. TANGUY	Maître de conférences, Université Paul Sabatier Toulouse, IMFT UMR 5502	Rapporteur
G. BALARAC	Professeur, Grenoble-INP ENSE3, LEGI UMR 5519	Examinateur
T. MÉNARD	Maître de conférences, Université de Rouen, CORIA-CNRS UMR 6614	Examinateur
S. POPINET	Directeur de recherches CNRS, Sorbonne Université, Institut Jean le Rond d'Alembert UMR 7190	Examinateur
R. MERCIER	Expert combustion, Safran Tech	Encadrant de thèse
V. MOUREAU	Chargé de recherches CNRS, CORIA-CNRS UMR 6614	Encadrant de thèse
A. BERLEMONT	Directeur de recherches CNRS, CORIA-CNRS UMR 6614	Directeur de thèse

Thèse dirigée par Alain BERLEMONT, encadrée par Vincent MOUREAU et Renaud MERCIER, laboratoire CORIA CNRS UMR6614 et Safran Tech

INSA INSTITUT NATIONAL DES SCIENCES APPLIQUÉES ROUEN NORMANDIE

ED PSIME

CORIA UMR 6614

Résumé : Simulation de l'atomisation primaire dans des injecteurs aéronautiques avec une méthode d'adaptation de maillage massivement parallèle

Dans les turbines à gaz aéronautiques, le système d'injection de kérosène liquide joue un rôle primordial sur l'opérabilité de la chambre de combustion. Il conditionne directement la stabilité de la flamme et les performances d'allumage, de réallumage et de non-extinction. De plus, le système d'injection doit avoir le comportement attendu sur une très large gamme de conditions opératoires : faible débit liquide au ralenti ou au réallumage / fort débit liquide au décollage, faible ou forte viscosité selon la température du carburant. La conception du système d'injection est donc difficile et résulte de plusieurs compromis. Cette tâche est rendue encore plus complexe par le manque de caractérisation expérimentale du système d'injection en conditions opératoires réalistes.

Cette thèse cherche à évaluer les capacités de la simulation instationnaire et plus particulièrement de la simulation aux grandes échelles (LES) pour la simulation de l'atomisation primaire et la prédiction des performances des systèmes d'injection de kérosène liquide dans les chambres de combustion aéronautiques. La simulation numérique présente l'avantage de donner accès à l'ensemble des quantités importantes pour mesurer ces performances : taille et volume des gouttes de carburant, distribution spatiale des gouttes, taux d'entraînement du gaz, ... Cependant, l'atomisation primaire étant un phénomène fortement non-linéaire, ce type de simulation requiert des résolutions de maillage très fines des couches de cisaillement et des interactions entre les deux phases fluides, afin de reproduire fidèlement la dynamique de l'écoulement. Cette nécessité augmente considérablement leur coût de calcul. Le caractère fortement multi-échelle de l'atomisation nécessite donc d'avoir recours à un grand nombre de processeurs en parallèle et à des méthodes numériques Eulériennes ou Lagrangiennes adaptées aux caractéristiques locales du spray, capables d'exploiter cette puissance de calcul.

Le logiciel YALES2, développé par le laboratoire CORIA et utilisé dans le groupe Safran, dispose d'un solveur pour la simulation de l'atomisation primaire basé sur la méthode Ghost-Fluid et sur le suivi d'interface de type Level-Set Conservative. Ce logiciel comporte également des méthodes de résolution des équations de Navier-Stokes à faible nombre de Mach efficaces sur un grand nombre de processeurs. Toutes ces méthodes combinées ont été utilisées préalablement à cette thèse pour simuler l'atomisation primaire en sortie d'un injecteur à triple disque et l'atomisation airblast à prefilming dans un injecteur réaliste fourni par Safran Helicopter Engines. Ces différentes études Eulériennes ont montré l'importance de la résolution du maillage sur la dynamique de l'interface. La prédiction de l'atomisation nécessite en effet de résoudre finement les modes instables de l'interface pour obtenir la formation des premiers ligaments et leur subséquente rupture en gouttelettes.

Pour faire face à ces défis, une méthode d'adaptation de maillage parallèle basée sur la librairie d'adaptation séquentielle libre MMG créée à l'INRIA, a été développée dans le logiciel YALES2. Cette méthode combine adaptation séquentielle et répartition de charge parallèle pour réaliser l'adaptation de maillage en parallèle sur des maillages de grande taille. Ce type de méthode d'adaptation peut maintenant servir en cours de simulation à concentrer les cellules du maillage où l'interface liquide/gaz se trouve afin de réduire le coût de calcul. En combinant cette adaptation de maillage isotrope au solveur Level Set et à un modèle de transport de spray Lagrangien, on peut obtenir un couplage cohérent des différentes approches en respectant leur domaine de validité.

L'objectif de cette thèse est de combiner l'ensemble de ces techniques pour modéliser et

simuler l'atomisation primaire de kérosène liquide avec le logiciel YALES2 à l'intérieur et en sortie des systèmes d'injection aéronautiques. La thèse se décompose en deux parties : (i) Le développement et la validation de méthodes numériques Eulériennes robustes, massivement parallèles, et précises pour l'atomisation turbulente sur maillages non-structurés adaptatifs; (ii) Le couplage de ces méthodes avec une approche Lagrangienne pour le transport du spray. Les simulations réalisées sont validées par la comparaison à des mesures expérimentales, et à des solutions théoriques ou numériques de référence.

Mots clés : Ecoulements diphasiques incompressibles; Atomisation primaire; Volumes Finis; Simulation aux grandes échelles; Maillages non structurés; Adaptation de maillage dynamique; Level set conservative; Calcul haute performance; Géométries complexes

***

Abstract: Numerical simulation of primary atomization in aeronautical injectors using a massively parallel adaptive mesh refinement technique

In aeronautical gas turbines, the liquid kerosene injection system has a crucial role on the combustor operability. Indeed, it directly conditions flame stability and ignition performances, as well as re-ignition and non-flameout. Moreover, the injection system must have the expected behavior on a broad range of operating conditions: low liquid flow rate at low velocity or at re-ignition/high liquid flow rate at take-off, variable fuel viscosity depending on its temperature. The design of injection systems is thus challenging and therefore the result from several compromises. This task is made even more complex by the lack of experimental characterization of injection systems in realistic operating conditions.

This doctoral thesis seeks to assess the capabilities of unsteady simulations, Large-Eddy Simulations (LES) in particular, for simulating primary atomization and predicting the performances of liquid kerosene injection systems in aeronautical combustion chambers. Numerical simulations allow to access a set of important parameters to measure these performances: size and volume of fuel drops, spatial distribution of drops, gas drive rate, ... However, primary atomization is a highly non-linear process, and hence this type of simulations requires very fine grid resolutions of shear layers and interactions between the two fluid phases, in order to reliably reproduce the complex flow dynamics. This requirement significantly increases their computational cost. Thus, the multi-scale nature of atomization necessitates to use a large number of processors in parallel, and to employ Eulerian or Lagrangian numerical methods suited for local spray characteristics, able to exploit this computational power.

The YALES2 software program, developed at CORIA laboratory and used in Safran group, features a solver for simulating primary atomization, based on the Ghost-Fluid Method and on the Accurate Conservative Level Set interface-capturing method. This software also features massively parallel numerical methods for solving the low-Mach Navier-Stokes equations. All these numerical methods combined have been employed prior to this doctoral thesis for simulating primary atomization of a liquid jet discharged from a triple-disk injector, and of the airblast atomization with prefilming in a realistic industrial injector provided by Safran Helicopter Engines. These various Eulerian studies have shown the importance of local mesh resolution on the interface dynamics. Indeed, numerically predicting atomization necessitates the fine resolution of unstable interfacial modes to yield the formation of the first ligaments and their subsequent break-up into small droplets.

To address these challenges, a parallel adaptive mesh refinement strategy based on the MMG free sequential adaptation library, created at INRIA, has been developed in the YALES2 software program. This method combines sequential adaptation together with parallel load balancing to perform parallel adaptive mesh refinement on large and massively-distributed grids. It can now be used throughout a simulation to concentrate the mesh cells at the liquid-gas interface location to reduce the computational cost. By combining this isotropic dynamic mesh adaptation strategy with the Level set solver and with a Lagrangian solver to model the transport of the dispersed spray, one can obtain a consistent coupling of the various approaches respecting their validity range.

The aim of this doctoral thesis is to combine these various techniques to model and compute liquid kerosene primary atomization using the YALES2 software program, within and at the outlet of aeronautical injection systems. The thesis is divided into two parts: (i) Development and validation of robust, massively parallel, and accurate Eulerian numerical methods for simulating turbulent atomization on adaptive unstructured grids; (ii) Coupling of these methods to a Lagrangian modeling approach for spray transport. The conducted simulations are validated against experiments, and against theoretical or numerical reference solutions.

Keywords: Multiphase flows; Primary atomization; Finite volume; Incompressible flow LES; Unstructured grids; Adaptive mesh refinement; Conservative level set; High-performance computing; Complex geometries

***

Remerciements

Voici venu le temps des remerciements. Je souhaiterais tout d'abord remercier les membres du jury d'avoir accepté d'examiner mon travail de doctorat, et pour la discussion lors de la soutenance, que j'ai sincèrement appréciée et qui fut très enrichissante. Je remercie également Alain Berlemont d'avoir accepté d'être mon directeur de thèse, et pour sa bienveillance lors de ces trois (que dis-je, quatre !) années de thèse.

Ensuite, je me dois d'exprimer ma gratitude envers mes encadrants de thèse : Vincent Moureau et Renaud Mercier, pour leur soutien technique et humain tout au long de la thèse, ainsi que pour leurs nombreux conseils. Ils ont su me pousser, à juste titre, et me motiver afin que ce travail soit le plus abouti possible, aussi bien d'un point de vue académique qu'industriel. Un grand merci à vous deux, et j'espère que l'on aura l'occasion de collaborer à nouveau dans le futur. J'adresse aussi mes sincères remerciements à Ghislain Lartigue et Pierre Bénard, les autres encandrants de l'équipe YALES2, pour leur aide précieuse et leur regard extérieur toujours pertinent.

Merci également à tous les collègues rencontrés au cours de ce périple : dans l'équipe YALES2, plus largement au CORIA, et à Safran Tech, qui ont rendu mon expérience du doctorat plus riche. Une mention spéciale à Carlos García-Guillamón, dont la soutenance sur un sujet étroitement lié au mien devrait suivre sous peu. Thanks as well to my new colleagues at Otto-von-Guericke-Universität Magdeburg, for their encouragement before the defense: I hope you enjoyed the live film screening!

Je me tourne ensuite vers ma famille et mes amis, qui ont su me soutenir et me faire penser à autre chose que la thèse lorsqu'il le fallait. Merci notamment au frangin, Guillaume, qui m'a aidé à répéter ma présentation de soutenance jusqu'à 2h du matin, et qui a démarré sa propre thèse en rayonnement des gaz : tu sais ce qui t'attend !

Enfin, merci à ma chère Mashenka, pour sa patience ("promis je relance juste mon calcul !") et pour son soutien sans faille, en particulier lors des moments difficiles de fin de thèse.

Contents

Introduction **5**

I Interface-capturing procedure on adaptive unstructured grids **5**

1 Computational Fluid Dynamics with interfaces **7**
1.1 Navier-Stokes equations with interfaces 7
1.1.1 General equations 7
1.1.2 Incompressible flows 8
1.1.3 Incompressible flows with surface tension 9
1.1.3.1 Molecular origin of surface tension 9
1.1.3.2 The one-fluid formulation 9
1.1.3.2.1 Volumetric formulation 10
1.1.3.2.2 Integral formulation 11
1.1.3.3 Formulation with jump conditions 11
1.1.4 Non-dimensional numbers 12
1.2 Numerical resolution of incompressible Navier-Stokes equations 13
1.2.1 Projection method 13
1.2.2 Turbulence modeling - Large-Eddy Simulation (LES) 15
1.2.3 Meshing 18
1.2.4 Finite-Volume discretization - Control volume 21
1.2.5 Time integration 24
1.2.6 Domain decomposition - Parallelization 25
1.3 Sharp interface-capturing techniques 26
1.3.1 Volume Of Fluid (VOF) 27
1.3.1.1 Algebraic VOF 28
1.3.1.1.1 Compressive schemes 29
1.3.1.1.2 THINC schemes 30
1.3.1.2 Geometric VOF 32
1.3.1.2.1 Interface reconstruction 32
1.3.1.2.2 Interface advection 33
1.3.2 Level Set (LS) 37
1.3.2.1 Standard Level Set (SLS) 37
1.3.2.2 Conservative Level Set (CLS) 40
1.3.3 Coupled Level-Set and Volume-Of-Fluid method (CLSVOF) 44
1.3.4 Moment Of Fluid methods (MOF) 44
1.4 Surface-tension modeling methods 46
1.4.1 Surface tension formulations 47
1.4.1.1 Continuum Surface Stress (CSS) - Integral formulations 47
1.4.1.2 Continuum Surface Force (CSF) 48
1.4.1.3 Smoothed interface representation 48
1.4.1.4 Ghost-Fluid Method (GFM) 48
1.4.2 Well-balanced schemes - Discrete Laplace balance 49

1.4.3 Interface curvature computation . . . 51
1.4.3.1 Implicit surface representation . . . 51
1.4.3.2 Explicit surface representation . . . 51
1.4.4 Explicit time integration - Capillary time-step restriction . . . 53

2 A two-phase flow solver for adaptive unstructured grids 55
2.1 Overview of the framework . . . 56
2.1.1 Accurate Conservative Level Set . . . 56
2.1.2 Navier-Stokes equations . . . 58
2.1.3 Summary of the full solution procedure . . . 58
2.2 Accurate Conservative Level Set algorithm . . . 59
2.2.1 Narrow-band ACLS . . . 59
2.2.2 Signed-distance function reconstruction: the GPMM algorithm . . . 59
2.2.2.1 Sorting of the node pairs . . . 60
2.2.2.2 Determination of intersection points . . . 60
2.2.2.3 Segmentation/Triangulation of the interface . . . 61
2.2.2.4 Sorting of markers and distance computation . . . 62
2.2.2.5 Optimization and parallelization . . . 63
2.2.2.6 Final nodal signed-distance computation in the narrow band . . . 63
2.2.3 Finite-volume framework for unstructured grids: spatial discretization and notations . . . 64
2.2.4 Interface normals and curvature computation . . . 65
2.2.4.1 Computation of curvature at nodes . . . 65
2.2.4.2 Interpolation for crossing pairs . . . 65
2.2.5 Computation of the interface profile thickness . . . 66
2.2.6 Discretization of the reinitialization equation . . . 66
2.2.7 Transport of the ACLS function - Time integration . . . 67
2.3 Incompressible flow solver . . . 69
2.3.1 Velocity prediction with momentum correction . . . 69
2.3.2 Ghost-Fluid Method . . . 70
2.3.2.1 Discretization of the pressure gradient . . . 70
2.3.2.2 Implementation in the Poisson solver . . . 71
2.3.3 Time integration - Capillary timestep restriction . . . 72
2.4 Isotropic dynamic mesh adaptation strategy . . . 72
2.4.1 Target metric generation . . . 73
2.4.2 Automatic triggering . . . 73
2.4.3 Parallel iterative remeshing . . . 74

3 Validation on unstructured grids 77
3.1 Accuracy of signed-distance and curvature computation . . . 77
3.2 Interface transport tests . . . 82
3.2.1 Static sphere on static tetrahedral grids . . . 82
3.2.2 Rotation of a notched disk on static triangular grids . . . 82
3.2.3 Circle deformation in a vortex on static triangular grids . . . 84
3.2.4 Sphere deformation in a vortex on adaptive tetrahedral grids . . . 86
3.3 Capillary-driven flows . . . 89
3.3.1 Damping of a capillary wave on static triangular grids . . . 89

3.3.2 Spurious currents on static tetrahedral grids . . . 92
3.3.3 Rayleigh-Plateau instability on static tetrahedral grids . . . 93
3.3.4 3D droplet collision using adaptive mesh refinement . . . 96

4 Application to injection systems 99
4.1 LES of a low-pressure water jet from a compound nozzle . . . 99
4.1.1 Internal flow . . . 100
4.1.2 External flow . . . 102
4.1.3 Computational performances . . . 102
4.2 LES of a high-pressure kerosene jet in crossflow . . . 106
4.2.1 Numerical setup . . . 106
4.2.2 Jet topology and breakup . . . 108
4.2.3 Vertical trajectory and experimental validation . . . 108
4.2.4 Computational performances . . . 112

II Multiscale modeling on adaptive unstructured grids 115

5 Lagrangian Particle Tracking for two-phase liquid-gas flows 117
5.1 Governing equations of the Euler-Lagrange framework . . . 117
5.1.1 Equations of motion of Lagrangian point particles . . . 117
5.1.2 One-way coupling approximation . . . 120
5.1.3 Influence of the particles on the flow: two-way coupling . . . 120
5.2 Conversion between Eulerian and Lagrangian frameworks . . . 122
5.2.1 Identification of isolated Eulerian liquid structures . . . 122
5.2.2 Conversion criteria between Eulerian and Lagrangian frameworks . . . 122
5.2.2.1 Eulerian-to-Lagrangian transfer . . . 122
5.2.2.1.1 Size criterion - Threshold volumes . . . 123
5.2.2.1.2 Shape criterion - Aspect ratio . . . 123
5.2.2.1.3 Distance criterion . . . 123
5.2.2.2 Lagrangian-to-Eulerian transfer . . . 124
5.2.3 Coupling with Adaptive Mesh Refinement . . . 125
5.2.4 Summary of the conversion procedure . . . 125

6 Euler-Lagrange simulations using adaptive unstructured grids 127
6.1 Settling droplet . . . 127
6.2 Accelerated droplet . . . 129
6.3 High-pressure kerosene jet in crossflow . . . 131
6.3.1 Visualization of the jet . . . 131
6.3.2 Computational performances . . . 136

Conclusion and future work 144

Long résumé 145

List of Figures

1 Representation of an aeronautical combustor, in which many physical phenomena take place: atomization, droplet transport, evaporation, turbulent mixing and combustion. . . . 1
2 Illustration of primary and secondary atomization regimes for a swirling spray, obtained in the experimental study of [2]. This figure is reprinted from [1]. 2

1.1 Interaction of liquid molecules, resulting in interfacial tension. 9
1.2 Fragment of *Studies of water passing obstacles and falling* by Leonardo da Vinci, c. 1508-1509 (Public Domain, from http://commons.wikipedia.org). This figure is reprinted from [60]. 15
1.3 Examples of multi-block structured meshes. 18
1.4 Examples of 2D unstructured meshes. Reprinted from [11]. 19
1.5 Jet created by the collapse under surface tension of an air bubble just below the surface of still water: the resolution varies along the interface according to local interface curvature. Reprinted from [75]. 19
1.6 Illustration of the skewness computation based on the equilateral volume for a triangular mesh cell. Reprinted from [76]. 20
1.7 Dynamic mesh adaptation using YALES2/MMG for combustion applications [79, 80]. The top slice depicts the temperature field T, while the bottom one shows the heat release $\dot{\omega}_T$ together with the adapted tetrahedral grid. 20
1.8 Notations for a staggered MAC grid. Reprinted from [49]. 22
1.9 Grid notations for the YALES2 flow solver; Node-centered control volume. . . . 22
1.10 Finite-volume pair-based discretization in YALES2, using node-centered control volumes V_i. 23
1.11 Double domain decomposition of a 2D triangular mesh. The cells located between black thick lines are those shared by different processors, and are thus involved in external communications (MPI). The grey cells are at the boundary of different cell groups within a processor (internal communications). Reprinted from [58]. . 25
1.12 Illustration of the Front-Tracking method. Reprinted from [89]. 26
1.13 Classification of sharp interface-capturing techniques. Inspired by [102]. 28
1.14 Illustration of upwind U, donor D, and acceptor A cells associated with a cell face f on a polyhedral mesh. U, D, and A are determined by the orientation of the local velocity field $\mathbf{u}_f$. Reprinted from [48]. 29
1.15 Illustration of the mappings used in the split advection schemes: (a) Contraction using a horizontal onto-cell implicit linear mapping $\Pi_{I,x}$; (b) Contraction using a horizontal out-of-cell explicit linear mapping $\Pi_{E,x}$. 35
1.16 Illustration of the flux computation in various cartesian unsplit advection schemes. This figure is reprinted from [136]. 37
1.17 Electrohydrodynamic-assisted atomization of a liquid jet. Reprinted from [137]. . 38
1.18 One-dimensional illustration of the standard level set method: (a) The level set is computed in the entire domain; (b) The level set is computed only in the neighborhood of the interface Γ, here with $k = 4$ points on each side of Γ (adapted from [92]). In this work: "Fluid 1" is gas, "Fluid 2" is liquid. 39

1.19 Turbulent atomization of a liquid Diesel jet with the original Accurate Conservative Level Set (ACLS) method (2008). Reprinted from [31]. 42
1.20 Outcome of a drop impact onto a static free surface; Left: uniform reinitialization, Right: Localized reinitialization. Reprinted from [143]. 43
1.21 DNS of a turbulent liquid jet injected into quiescent gas using the reformulated ACLS method (2017): renderings of the liquid-gas interface. Reprinted from [35]. 43
1.22 Development of an atomizing liquid jet using the CLSVOF method, implemented in the Archer DNS code. Reprinted from [146]. 44
1.23 Illustration of the MOF-PLIC interface reconstruction using liquid centroids. Inspired by [152]. 45
1.24 MOF-CLSVOF hybridation technique applied to the atomization of a turbulent liquid diesel jet. Red regions are treated using the MOF interface reconstruction, while blue regions employ the CLSVOF method. Reprinted from [153]. 46
1.25 Spurious currents around a stationary bubble using a VOF method. Reprinted from [88]. 50

2.1 Interface capturing on a one-dimensional example using the ACLS method. $x_\Gamma = 0$; $\delta_\psi = 4\varepsilon$. 56
2.2 (a) Flagging of the narrow band around the interface Γ, here ik is a pair of nodes crossing the interface; (b) Pair-based band-layer index B_{ik} for data propagation in the Geometric-Projection Marker Method (GPMM). 59
2.3 Segmentation of the interface in 2D. $P_K(M_I)$ is the K-th intersection point associated with the I-th marker M_I. 61
2.4 Projection algorithm to compute the exact geometric absolute distance d to the triangulated interface Γ^*. 62
2.5 Finite-volume pair-based discretization in YALES2, using node-centered control volumes V_i. 64
2.6 Full ACLS procedure: the blue boxes represent computations in the narrow band. 68
2.7 Ghost-Fluid Method: extension of the jump of a physical quantity ζ. 71
2.8 Adaptive mesh refinement strategy: refinement based on the distance $\mathcal{D}$ to the interface. 73
2.9 1D illustration of the AMR triggering using a user-defined protected distance $|\phi_p| = (N_p/2)\Delta x_{\min}$: (a) Liquid-gas interface Γ right after an adaptation process (instant t); (b) Interface N fluid iterations later (instant $t + N\Delta t$), triggering of AMR; (c) Remeshing around Γ at $t + N\Delta t$. 75
2.10 Bad-quality grid cells at interfaces between processors. Courtesy of A. Froehly - ParMMG [77]. 76
2.11 Schematic of the two-level moving interface parallel adaptation strategy for 4 processors. The thin black lines represent the interfaces between cell groups and the different colors correspond to the processor ranks. Reprinted from [80]. 76

3.1 Static droplet of radius R_d in a squared domain. The signed distance from a node i to the interface Γ is ϕ_i. κ_Γ is the curvature at the interface. 78
3.2 L_2 and L_∞ of the reconstructed signed-distance function ϕ with the 2D GPMM algorithm (Geometric-Projection Marker Method), in first (a) and sixth (b) narrow-band levels $|b|$. The dashed and solid lines are first and second-order convergence, respectively. 79

3.3 L_2 and L_∞ of the reconstructed signed-distance function ϕ with the 3D GPMM algorithm (Geometric-Projection Marker Method), in first (a) and sixth (b) narrow-band levels $|b|$. The dashed and solid lines are first and second-order convergence, respectively. 79
3.4 L_2 and L_∞ norms of the 2D (a) and 3D (b) interface curvature κ_Γ, considering two spatial discretizations of the Hessian operator: with compact and non-compact stencils. The dashed and solid lines are first and second-order convergence, respectively. 81
3.5 L_2 and L_∞ norms of the 3D interface curvature κ_Γ, using Goldman's formula and the classical formula, both discretized with non-compact stencils. The dashed and solid lines are first and second-order convergence, respectively. 81
3.6 3D static sphere transport test on two tetrahedral grids: comparison between Eq. (2.5) (original ACLS reinitialization) and Eq. (2.7) (reformulated reinitialization) after 50,000 iterations of reinitialization. 82
3.7 Interface contours after 10 rotations for various mesh resolutions. The thin line represents a simulation performed with 400 points in the disk radius. 83
3.8 Scalar fields after 10 rotations with $R/\Delta x = 50$. (a) Signed narrow-band levels b; (b) Reconstructed signed-distance function ϕ_{GPMM}. Black contours represent the interface. 83
3.9 Node normal vector field $\mathbf{n}_i = \boldsymbol{\nabla}\phi_{\mathrm{GPMM}}\Big|_i^{\mathrm{4th}}$ around the interface after 10 rotations with $R/\Delta x = 50$, colored by the signed narrow-band levels b. 84
3.10 Shape error E_{shape} after 10 rotations of the 2D notched disk ($t = T$), for various mesh resolutions: from 12.5 to 400 points in the disk radius. The dashed and solid lines are first and second-order convergence, respectively. 85
3.11 Interface contours at maximum deformation ($t = T/2$; top) and final shape ($t = T$; bottom) for various mesh resolutions. The thin line represents a simulation performed with 400 points in the disk radius. 86
3.12 Shape error E_{shape} at the end of the 2D circle deformation in a vortex test ($t = T$), for various mesh resolutions: from 12.5 to 400 points in the disk radius. The dashed and solid lines are first and second-order convergence, respectively. . . . 87
3.13 Interface contours at maximum stretch ($t = T/2$; top) and final shape ($t = T$; bottom) for various adaptive mesh resolutions. 88
3.14 Tetrahedral cell size distribution at maximum stretch ($t = T/2$) for various adaptive mesh resolutions. 88
3.15 Interface contours at maximum stretch ($t = T/2$) and final shape ($t = T$) for various uniform mesh resolutions. 89
3.16 Illustration of the damped wave test case. 90
3.17 (a) Temporal evolution of the dimensionless wave amplitude A/λ over non-dimensional time $t^* = \omega_0 t$; (b) Instantaneous relative amplitude error $|A - A_{\mathrm{th}}|/A_0$. 91
3.18 L_2 relative error norm for the present study, compared to the CSF method [154] and Popinet 2009 [21]. 92
3.19 Spurious currents around a static droplet on tetrahedral meshes. $\mathrm{La} = 2,000,000$. $\rho_l/\rho_g = 1000$. $\mu_l/\mu_g = 50$. 93
3.20 Rayleigh-Plateau instability on a homogeneous tetrahedral mesh: interface contours at various non-dimensional times t/t_σ. $\mathrm{La} = 18,530$. $\xi = kr_0 = 0.52$. $r_0/\Delta x = 6$. $t_\sigma = 1/\omega_0$. 95

3.21 Interface contour in the pinch-off region. $\mathrm{La} = 18,530$. $\xi = kr_0 = 0.52$. (a) $r_0/\Delta x = 6$; (b) $r_0/\Delta x = 24$. 95
3.22 Rayleigh-Plateau instability. $\mathrm{La} = 18,530$. $r_0/\Delta x = 6$. (a) Temporal evolution of the relative deformation based on the maximum radius for $\xi = kr_0 = 0.52$, $t_\sigma = 1/\omega_0$. (b) Dispersion diagram; the error bars represent the standard deviation SD of the growth-rate signal $\pm$SD (see Eq. (3.22)). 96
3.23 Water/Air droplet collision. $\mathrm{We} = 23$; $\mathrm{Oh} = 0.0047$; X=0; $\Delta = 1$; $D/\Delta x_{\min} = 133$. 97
3.24 Water/Air droplet collision simulation compared to the experiment of [179], performed in the same physical conditions. $\mathrm{We} = 23$; $\mathrm{Oh} = 0.0047$; X=0; $\Delta = 1$; $D/\Delta x_{\min} = 133$. 98

4.1 Compound nozzle geometry: triple-disk injector. 99
4.2 Illustration of the double-vortex internal flow in the cavity disk (Disk 2 in Fig. 4.1): velocity magnitude along with streamlines. 100
4.3 y^+ distribution represented at boundaries. $d_{\mathrm{inj}} = 180\mu\mathrm{m}$. $\mathrm{Re} = 3653$. $\mathrm{We_l} = 1061$. 101
4.4 Velocity statistics: (a) Mean (Eq. (4.1)) and (b) Root Mean Square (Eq. (4.2)) velocity projected in the exit section. $d_{\mathrm{inj}} = 180\mu\mathrm{m}$. $\mathrm{Re} = 3653$. 101
4.5 Interface contours and Q-criterion of the water jet atomizing from a low-pressure compound nozzle in quiescent air at the two investigated resolutions: (a) $\Delta x_{\min} = 5\mu\mathrm{m}$ and (b) $\Delta x_{\min} = 2.5\mu\mathrm{m}$. 103
4.6 Closeup on some capillary effects during the atomization process of a water jet in quiescent air from a low-pressure compound nozzle. $d_{\mathrm{inj}} = 180\mu\mathrm{m}$. $\mathrm{Re} = 3653$. $\mathrm{We_l} = 1061$. $\Delta x_{\min} = 2.5\mu\mathrm{m}$. 104
4.7 Qualitative comparison of the jet topology close to the nozzle between (a) the experiment, (b) the present ACLS/LES study using YALES2 ($\Delta x = 2.5\mu\mathrm{m}$ at the interface), and (c) the CLSVOF/DNS performed with the Archer code ($\Delta x = 1.44\mu\mathrm{m}$). This figure has been built upon the one of [40]. 105
4.8 Mid-plane colored by cell size. $d_{\mathrm{inj}} = 180\mu\mathrm{m}$. (a) $\Delta x_{\min} = 5\mu\mathrm{m}$; (b) $\Delta x_{\min} = 2.5\mu\mathrm{m}$. 105
4.9 Estimation of the spray angle α at different distances z/d_{inj} from the nozzle exit. $d_{\mathrm{inj}}/\Delta x_{\min} = 72$. 106
4.10 Numerical domain and boundary conditions of the experimental test bench of Becker and Hassa [41]: (a) complete domain, (b) detailed view of the injection nozzle. All dimensions are in mm. 107
4.11 Lateral view of meshes and interface contours near the injector at time instant $t = 0.3$ms. $\mathrm{q} = 6$. $\mathrm{We}_g = 1470$. 109
4.12 Instantaneous front (left) and lateral (right) views at time instant $t = 0.3$ms for the two operating conditions. $d_{\mathrm{inj}}/\Delta x_{\min} = 45$. $\mathrm{q} = 6$. $\mathrm{We}_g = 830$ (top), $\mathrm{We}_g = 1470$ (bottom). 110
4.13 Numerical mean jet trajectories for (a) $\mathrm{We}_g = 830$ and (b) $\mathrm{We}_g = 1470$ compared with the experimental correlation from Becker [41]. The grey area represents the standard deviation of the experimental trajectory correlation provided by [41]. . . 111

5.1 Illustration of the distance criterion, using the influence region (delimited by the black dashed line) of the equivalent spherical droplet (black solid line) associated with the resolved liquid structure of interest (blue solid line). The resolved droplet is represented as a large ellipsoid to ease the reading of the figure. 124

5.2 Flowchart of the Eulerian-to-Lagrangian conversion. 126

6.1 Numerical setup for the settling droplet test case: (a) At the initialization; (b) At the end of the first iteration. The droplet falls along the $y < 0$ direction. 128
6.2 (a) Temporal evolution of the y-position of the falling droplet; (b) Temporal evolution of the u_y-velocity of the falling droplet against the terminal velocity for unbounded domains of Clift and Gauvin (1970) [201]. 129
6.3 Numerical setup for the accelerated droplet test case (LPP simulation): (a) At the initialization; (b) At the end of the first iteration. The droplet is accelerated along the $x > 0$ direction. 130
6.4 (a) Temporal evolution of the x-position of the accelerated droplet; (b) Temporal evolution of the u_x-velocity of the accelerated droplet. 131
6.5 Topology of the high-pressure kerosene jet in crossflow at $t = 0.3\mathrm{ms}$, using the hybrid Euler-Lagrange strategy versus the fully Eulerian simulation of [180]. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. 132
6.6 Topology of the high-pressure kerosene jet in crossflow and count of Lagrangian particles at $t = 0.375\mathrm{ms}$, using the hybrid Euler-Lagrange strategy. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. 133
6.7 Topology of the high-pressure kerosene jet in crossflow along with the adaptive tetrahedral grid at $t = 0.375\mathrm{ms}$, using the hybrid Euler-Lagrange strategy. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. 133
6.8 Temporal evolution of the high-pressure kerosene jet in crossflow along with the adaptive tetrahedral grid, using the hybrid Euler-Lagrange strategy. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. 134
6.9 Particle diameter distribution at $t = 0.375\mathrm{ms}$, using the hybrid Euler-Lagrange strategy. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. . . 135
6.10 Early stage of the development of the Eulerian-Lagrangian high-pressure kerosene jet in crossflow along with (a) the adaptive tetrahedral grid, and (b) the velocity field in the mid-plane. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. 136
6.11 High-pressure kerosene jet in crossflow - Closeup on an Euler-to-Lagrange conversion. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$. . . 136
6.12 Volume rendering of the atomizing jet issued from a low-pressure compound nozzle. 139
6.13 Isotropic triangular mesh (Left) versus Anisotropic triangular mesh (Right). Reprinted from [207]. 140
6.14 Liquid kerosene jet in crossflow using the multi-scale Euler-Lagrange strategy. . 141

List of Tables

2.1 A 2D marker list for a node i of the narrow band: 2 identificators $id_{1,2}$, 4 coordinates ($K = 1, 2$), 1 distance d. 61

3.1 Error norms of the reconstructed signed-distance function ϕ using the 3D GPMM algorithm, in first and sixth band levels for each resolution. The corresponding order of convergence are given within parentheses. 80
3.2 Error norms for each mesh resolution at the end of the notched disk test, and computational cost of the runs. 85
3.3 Error norms at the end of the circle deformation test, and computational cost of the runs. 87
3.4 Computational costs of the sphere deformation simulations using adaptive tetrahedral grids 88
3.5 Computational costs of the sphere deformation simulations using uniform tetrahedral grids 89
3.6 Physical and numerical parameters for the viscous damping of a capillary wave case. 91
3.7 Error levels of the L_2 norm (Eq. (3.15)) and reduced computational time (RCT; Eq. (3.8)) for each resolution. The corresponding order of convergence are given within parentheses. 91
3.8 Physical and numerical parameters for the measurements of spurious currents. . . 93
3.9 Cases definition for the Rayleigh-Plateau instability simulations. 94
3.10 Physical parameters for the Rayleigh-Plateau instability simulations. 94
3.11 Numerically predicted mean growth rates versus theoretical growth rates from Weber's theory, with $r_0/\Delta x = 6$. 96
3.12 Physical parameters for the water/air head-on droplet collision simulation. 97
3.13 Computational performances of the water/air head-on droplet collision simulation. 97

4.1 Computational costs and performances of the low-pressure compound injector simulations on adaptive grids. The simulations have been conducted on 2nd Gen AMD EPYC processors. 102
4.2 JICF operating points 107
4.3 Error levels of the L_2 norm (Eq. (4.6)) for jet in crossflow simulations 112
4.4 Number of computational cores, mesh cells and droplets generated for jet in crossflow simulations. All values correspond to time instant $t = 0.3$ms. 113
4.5 Computational performances for jet in crossflow simulations at time instant $t = 0.3$ms. 113

6.1 Physical parameters for the settling droplet test case. 128
6.2 Terminal velocity result and comparison for the settling droplet test case. 129
6.3 Physical parameters for the accelerated droplet test case. 129
6.4 Final axial positions x_f, axial velocities $\mathrm{u}_{x,f}$, and number of grid cells $N_{\mathrm{cells},f}$ for the accelerated droplet test case. 131

6.5 Number of computational cores, mesh cells and droplets generated for the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180], whose performance values are displayed in Chapter 4, Table 4.4. All values correspond to time instant $t = 0.3$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 22.5$). 137
6.6 Computational performances of the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180], whose performance values are displayed in Chapter 4, Table 4.5. All values correspond to time instant $t = 0.3$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 22.5$). . . 137
6.7 Number of computational cores, mesh cells and droplets generated for the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180]. All values correspond to time instant $t = 0.15$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 45$). 138
6.8 Computational performances of the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180]. All values correspond to time instant $t = 0.15$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 45$). 138

Introduction

Multiphase flows are ubiquitous in nature and in industrial systems, and understanding the formation of sprays is crucial for designing complex injection technologies. More specifically, turbulent atomization is of first importance in aeronautical combustors, into which fuel is injected in liquid form, goes under atomization, evaporation, turbulent mixing and eventually combustion (see Fig. 1). All these physical processes are strongly interdependent, and thus improving numerical

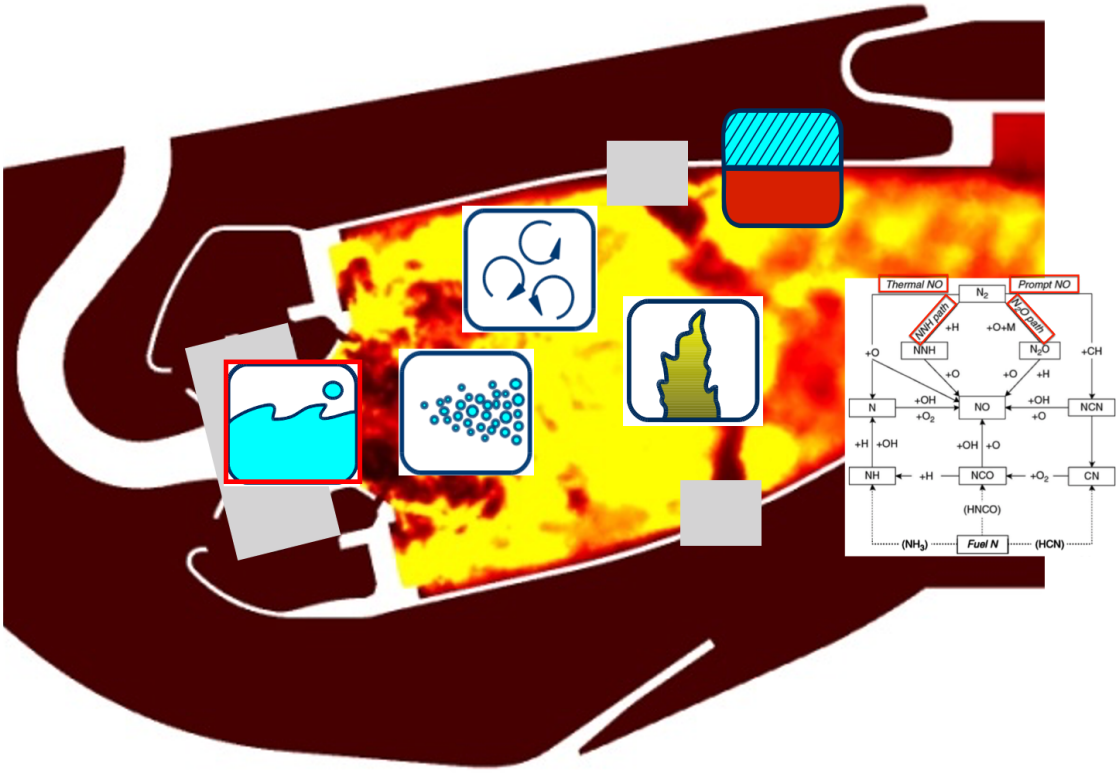
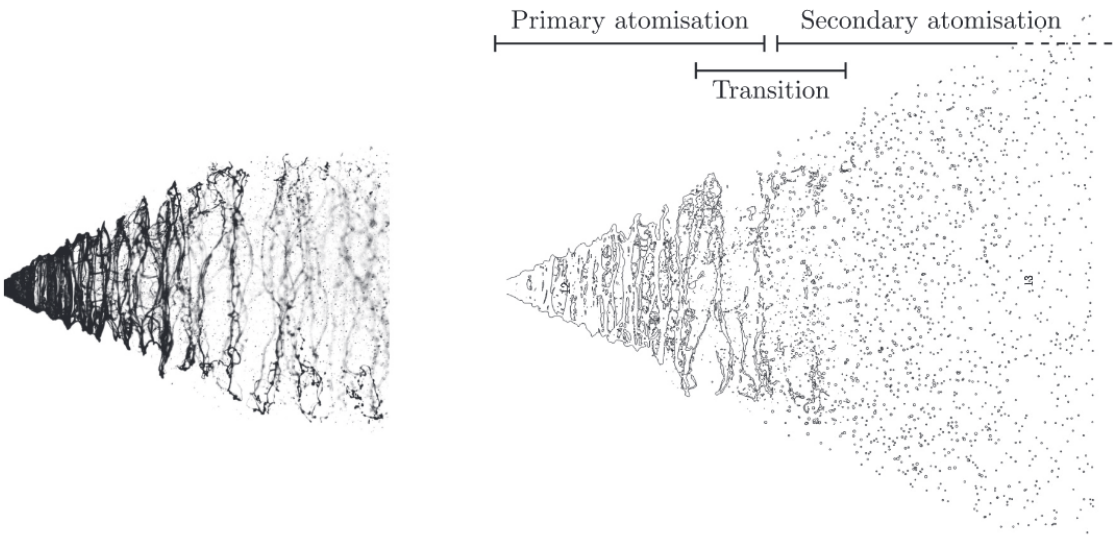


Figure 1: Representation of an aeronautical combustor, in which many physical phenomena take place: atomization, droplet transport, evaporation, turbulent mixing and combustion.

predictions of pollutant formation requires deep numerical insights into the atomization process. This process designates the evolution of large liquid structures into small droplets and thus governs the liquid droplet distribution in the combustor and the subsequent evaporation. The structure of an atomizing spray can be decomposed into two main regions: primary and secondary atomization regimes, as depicted in Fig. 2 for a swirling spray (reprinted from [1]).

In order to properly simulate an atomizing spray, the temporal evolution of the liquid-gas interface topology together with surface tension effects need to be accurately represented on a numerical grid. To this aim, interface-capturing techniques are classicaly employed, such as Volume Of Fluid (VOF) [3] or Level Set (LS) methods [4, 5], and are coupled to a numerical surface tension model. Realistic industrial injector geometries for primary atomization simulations are very complex, including sharp edges and narrow corners, which hinders the meshing task with cartesian grids: the use of unstructured meshes is thus well suited for this type of simulations [6]. Although the literature remains limited about unstructured atomization computations, previous



(a) Swirl injector atomisation experiment

(b) Schematic view of the spray and associated atomisation regimes

Figure 2: Illustration of primary and secondary atomization regimes for a swirling spray, obtained in the experimental study of [2]. This figure is reprinted from [1].

studies on static meshes can be found, such as [7, 8, 9, 10, 11, 12, 13, 14] for Volume Of Fluid (VOF) interface-capturing techniques, or [15, 16, 17, 18, 19] for Level Set methods (LS).

Despite many years of study, high-performance atomization simulations remain a challenge: the wide range of involved time and space scales leads to important calculation costs. To tackle this issue, the use of Adaptive Mesh Refinement (AMR) is particularly helpful, allowing the allocation of computational resources to the interface dynamics. A popular technique for solving incompressible two-phase flows on adaptive cartesian grids is tree-based AMR (quad/octree), as in the work of Popinet [20, 21]. Regarding unstructured meshes, Dai and Schmidt [22] proposed a moving-mesh algorithm to simulate free-surface flows with large deformations on adaptive tetrahedral grids. Both studies are proposed in the context of VOF methods, and the meshes are typically refined based on the local curvature of the interface. For classical Level Set methods on adaptive tetrahedral grids, Morgan and Waltz [23] introduced a method to treat evolving fronts with specified velocity, while Rodriguez et al. presented a parallel two-phase flow solver using a finite-element formulation in [24], where the adaptive mesh is refined based on the distance to the interface.

However, even with the help of AMR techniques, interfacial liquid-gas flows of practical interest may still exhibit prohibitive computational costs, due to the substantially large number of droplets formed during the atomization process, that need to be resolved in the numerical domain using the adaptive grid. Thus, to further increase the efficiency and the accuracy of the computations of such flows, especially the dispersed phase where only isolated drops are present, hybrid Eulerian-Lagrangian techniques have arisen over the past decade such as [25, 26, 27, 28, 1], all for cartesian grids. The idea is to convert the small and spherical resolved Eulerian droplets to Lagrangian Point Particles (LPP). With such multi-scale modeling strategy, the converted drops, usually poorly resolved by the grid, are treated more efficiently and accurately, and do not need to be resolved anymore: local coarsening of the mesh can thus be performed dynamically when using AMR [26, 28]. As a consequence, a mist of drops can be transported much further in a numerical domain, without adding a large number of mesh cells which would be required for transporting

many resolved drops over a large distance.

In this work, the Accurate Conservative Level Set (ACLS) method is used to capture the interface on 2D and 3D unstructured grids (triangular and tetrahedral): this method accurately predicts the interface dynamics while conserving a smoothed phase indicator. The ACLS method is thus not strictly mass-conserving, but the errors in the liquid mass conservation are small and controlled [29, 30, 31]. The ACLS method relies on the use of a hyperbolic tangent function to represent the interface, as in the THINC scheme of Xiao et al. [32, 33], proposed earlier in the context of algebraic VOF methods. The ACLS hyperbolic tangent function is advected by the fluid, and then reshaped using a reinitialization equation. ACLS on unstructured grids has been proposed before in [16, 17] on static meshes, and by Antepara et al. [34] for adaptive tetrahedral grids. In this latter, non-conforming AMR with an octree data structure is used. In the present work, massively distributed adaptive conforming unstructured grids are employed, and adapted isotropically using both metric-based and skewness-based quality criteria. We extend these previous unstructured ACLS studies by including an adapted and improved reinitialization method, first introduced by Chiodi and Desjardins for cartesian grids [35] and that has never been implemented to date for unstructured grids. In particular, interface-merging properties of this latest reinitialization equation are discussed. Indeed, the original ACLS equation involves numerical estimation of the hyperbolic tangent gradient, which is difficult to compute accurately on unstructured meshes. It is thus susceptible to induce artificial deformation of the interface. The latest equation takes advantage of a mapping onto a classical distance level set while much better preserving the interface shape. Moreover, in the proposed framework, a narrow band around the liquid-gas interface is built to compute interface features only in the area of interest [36], allowing significant computational savings. In order to compute interface normals and curvature, the signed-distance function is reconstructed in parallel at nodes in the narrow band using a Geometric-Projection Marker Method (GPMM). This method relies on the triangulation of the level set iso-contour and exact geometric projection to the closest surface elements. Surface tension is treated explicitly using the Ghost-Fluid Method (GFM) [37]. The interface-capturing procedure using adaptive unstructured grids is then coupled to a Lagrangian modeling strategy, designed for tetrahedral meshes: the small and spherical resolved droplets are converted to Lagrangian Point Particles (LPP), using conversion criteria based on numerical and geometrical considerations. The reverse transformation, namely the Lagrangian-to-Eulerian conversion, necessitates the implementation of a re-impact algorithm of a Lagrangian particle onto the resolved interface. Such algorithm is not yet implemented in the present multiscale procedure. As a consequence, only the droplets at a certain distance from the resolved interface will be converted. Once an Eulerian resolved droplet is transformed, the tetrahedral mesh is automatically and locally coarsen around the newly introduced Lagrangian particle.

This manuscript is divided into two major parts. The first part focuses on the Eulerian solver. After a first general chapter about incompressible two-phase liquid-gas flow modeling, the various building blocks of the procedure implemented in the YALES2 unstructured incompressible code [38] are presented, namely the ACLS interface-capturing technique, the fractional-step integration of Navier-Stokes equations with explicit treatment of jump conditions, and the dynamic mesh adaptation strategy. The accuracy, spatial convergence, and computational efficiency of the unstructured ACLS method are assessed on classical interface transport test cases. The behavior of the overall algorithm in the case of canonical capillary-driven two-phase flows is then verified. Finally, two LES simulations with large density ratios in complex geometries are performed on adaptive tetrahedral meshes: the break-up of a turbulent liquid jet discharged from a low-pressure compound nozzle [39, 40] and of a non-reactive high-pressure liquid jet in crossflow are shown at

various resolutions, demonstrating the accessibility of jet instabilities with 3D adaptive unstructured meshes. Results are validated against experiments performed in the same conditions [41, 42]. The second part of this manuscript presents the hybrid Eulerian-Lagrangian multiscale modeling strategy, designed for adaptive tetrahedral grids. The equations solved by the Lagrangian solver are first derived, and the underlying assumptions explicitly mentioned. Then, the conversion procedure between the two frameworks and its coupling with Adaptive Mesh Refinement (AMR) are detailed. The hybrid strategy is eventually applied and tested on canonical cases, and on the same high-pressure liquid jet in crossflow than the one presented in the first part of this work. It thus allows to assess the computational performances of the proposed multi-scale coupling procedure using massively-distributed adaptive unstructured grids.

Part I

Interface-capturing procedure on adaptive unstructured grids

CHAPTER 1

Computational Fluid Dynamics with interfaces

Contents

1.1 Navier-Stokes equations with interfaces 7
1.1.1 General equations 7
1.1.2 Incompressible flows 8
1.1.3 Incompressible flows with surface tension 9
1.1.4 Non-dimensional numbers 12
1.2 Numerical resolution of incompressible Navier-Stokes equations 13
1.2.1 Projection method 13
1.2.2 Turbulence modeling - Large-Eddy Simulation (LES) 15
1.2.3 Meshing 18
1.2.4 Finite-Volume discretization - Control volume 21
1.2.5 Time integration 24
1.2.6 Domain decomposition - Parallelization 25
1.3 Sharp interface-capturing techniques 26
1.3.1 Volume Of Fluid (VOF) 27
1.3.2 Level Set (LS) 37
1.3.3 Coupled Level-Set and Volume-Of-Fluid method (CLSVOF) 44
1.3.4 Moment Of Fluid methods (MOF) 44
1.4 Surface-tension modeling methods 46
1.4.1 Surface tension formulations 47
1.4.2 Well-balanced schemes - Discrete Laplace balance 49
1.4.3 Interface curvature computation 51
1.4.4 Explicit time integration - Capillary time-step restriction 53

1.1 Navier-Stokes equations with interfaces

1.1.1 General equations

Fluid flows are ubiquitous in nature and in industrial systems. A few examples include air flowing around an aircraft (aerodynamics), blood flows (hemodynamics), or ocean breaking waves. The

motion of fluids under the continuum hypothesis obey to the well-know Navier-Stokes equations. These equations are written in conservative form as:

$$\frac{\partial \rho}{\partial t} + \boldsymbol{\nabla} \cdot (\rho \mathbf{u}) = 0 \tag{1.1}$$

$$\underbrace{\frac{\partial (\rho \mathbf{u})}{\partial t}}_{\text{local momentum variation}} + \underbrace{\boldsymbol{\nabla} \cdot (\rho \mathbf{u} \otimes \mathbf{u})}_{\text{momentum convection}} = \underbrace{-\boldsymbol{\nabla} p}_{\text{pressure forces}} + \underbrace{\boldsymbol{\nabla} \cdot \boldsymbol{\tau}}_{\text{viscous forces}} + \underbrace{\rho \mathbf{g}}_{\text{gravitational forces}} \tag{1.2}$$

with ρ the fluid density, $\mathbf{u}$ the fluid velocity vector field, p the pressure field, $\boldsymbol{\tau}$ the viscous stress tensor, and $\mathbf{g}$ the gravitational acceleration. Eqs. (1.1) and (1.2) are mass and momentum conservation equations, respectively. For an isotropic Newtonian fluid, the viscous stress tensor is related to the strain rate tensor as:

$$\boldsymbol{\tau} = 2\mu \mathbf{S} + \lambda (\boldsymbol{\nabla} \cdot \mathbf{u}) \mathbf{I} \tag{1.3}$$

where μ is the fluid dynamic viscosity, $\mathbf{S} = 1/2 \left(\boldsymbol{\nabla} \mathbf{u} + (\boldsymbol{\nabla} \mathbf{u})^T \right)$ is the strain rate tensor, $\mathbf{I}$ is the identity matrix, and λ is the second viscosity coefficient. In the case of an anisotropic fluid, a viscosity tensor $\boldsymbol{\mu}$ would be necessary: this case is not considered in this work. The expression of the stress tensor $\boldsymbol{\tau}$ can be further simplified. Let us consider the volume viscosity μ_V of an isotropic Newtonian fluid, defined as $\mu_V = \lambda + (2/3)\mu$. Under Stokes' hypothesis, the dilatational viscosity μ_V is ignored:

$$\mu_V = \lambda + (2/3)\mu = 0 \tag{1.4}$$

leading to $\lambda = -(2/3)\mu$. the viscous stress tensor thus writes:

$$\boldsymbol{\tau} = \mu \left(\boldsymbol{\nabla} \mathbf{u} + (\boldsymbol{\nabla} \mathbf{u})^T \right) - \frac{2}{3} \mu (\boldsymbol{\nabla} \cdot \mathbf{u}) \mathbf{I} \tag{1.5}$$

1.1.2 Incompressible flows

The Mach number M, characterizes the flow velocity $|\mathbf{u}|$ relatively to the local speed of acoustic perturbations a:

$$\mathrm{M} = \frac{|\mathbf{u}|}{a} \tag{1.6}$$

with a defined as:

$$a = \sqrt{\left(\frac{\partial p}{\partial \rho} \right)_S} \tag{1.7}$$

The derivative is taken isentropically, i.e. at constant entropy S. Indeed, the propagation of an acoustic wave is approximated as an adiabatic process. In this work, the flows are all assumed to be incompressible, i.e. $\mathrm{M} \ll 1$. In this case, relative density variations are neglected [43]:

$$\frac{\Delta \rho}{\rho} \sim \mathrm{M}^2 \ll 1 \tag{1.8}$$

and hence density is taken as a constant. In these conditions, the continuity equation Eq. (1.1) reduces to:

$$\boxed{\boldsymbol{\nabla} \cdot \mathbf{u} = 0} \tag{1.9}$$

meaning that the velocity field is solenoidal. Please note that while only incompressible flows will be considered, Eq. (1.1) will be retained for numerical implementation reasons, in addition to Eq. (1.9). Finally, as a consequence of Eq. (1.9), the viscous stress tensor Eq. (1.5) can be simplified:

$$\boxed{\boldsymbol{\tau} = \mu \left(\boldsymbol{\nabla} \mathbf{u} + (\boldsymbol{\nabla} \mathbf{u})^T \right)} \tag{1.10}$$

1.1.3 Incompressible flows with surface tension

By considering an interfacial flow, involving two immiscible fluid phases (e.g. a liquid-gas flow), it is necessary to include a term accounting for the interface between the phases. In the case of liquid-gas flows, this term needs to include surface tension effects, and appears, explicitly or within the pressure term, in the momentum equation Eq. (1.2).

1.1.3.1 Molecular origin of surface tension

Let us consider a liquid-gas interface Γ. The molecules in the bulk liquid phase are pulled equally in every direction due to cohesive forces: the net force is hence zero. However, at the interface, the liquid molecules interact not only with other liquid molecules, but also with gaseous ones. Interaction forces between two molecules of the same type are called "cohesive forces", whereas interaction forces between two different types of molecules are "adhesive forces", see Fig. 1.1. This interfacial state results in a contraction of the surface to its minimum area.

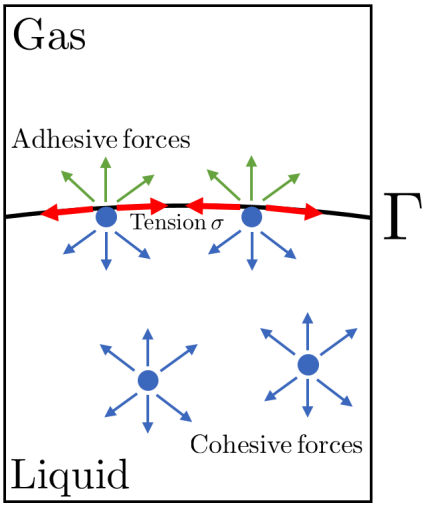


Figure 1.1: Interaction of liquid molecules, resulting in interfacial tension.

In terms of energy, liquid molecules at the surface have less bonds on average than the ones in the bulk of liquid. As a consequence, they are in a higher state of energy [44, 45, 46, 47]. This excess of surface energy can be expressed using the interfacial free energy E per unit area [48]:

$$\sigma = \frac{\partial E}{\partial A} \tag{1.11}$$

where σ is called the surface tension coefficient. This macroscopic quantity can be seen either as an energy per unit area, or equivalently as a force per unit length. For the liquid to minimize its energy state (i.e. reach a thermodynamic optimum), the interface will reduce to the minimum area with the smoothest shape possible. This is why small droplets are spherical: it minimizes surface energy.

1.1.3.2 The one-fluid formulation

In this formulation, the surface tension forces per unit volume $\mathbf{F}_\sigma$ are included explicitly in the momentum equation Eq. (1.2) and thus one set of governing equations is used for the whole flow

domain [49]:

$$\boxed{\frac{\partial(\rho\mathbf{u})}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = \underbrace{-\boldsymbol{\nabla}p}_{\text{pressure forces}} + \underbrace{\boldsymbol{\nabla}\cdot\boldsymbol{\tau}}_{\text{viscous forces}} + \underbrace{\rho\mathbf{g}}_{\text{gravitational forces}} + \underbrace{\mathbf{F}_\sigma}_{\text{surface tension forces}}} \tag{1.12}$$

The capillary forces $\mathbf{F}_\sigma$ have a role only at the interface. Away from it, the classical incompressible momentum equation is valid. In this subsection, two surface-tension formulations are presented: namely the volumetric and integral formulations, originally named and presented in [50].

1.1.3.2.1 Volumetric formulation

Let us consider an interface Γ between two phases in a fluid domain, and A the portion of the interface in a control volume V. The surface tension force by unit area $\mathbf{f}_\sigma$, i.e. the surface tension stress, is given by the stress balance equation [49]:

$$\mathbf{f}_\sigma = \underbrace{\sigma\kappa\mathbf{n}}_{\text{normal stress}} + \underbrace{\boldsymbol{\nabla}_A\sigma}_{\text{tangential stress}} \tag{1.13}$$

where σ is the surface tension coefficient, $\kappa = -\boldsymbol{\nabla}\cdot\mathbf{n}$ is the interface curvature, $\mathbf{n}$ is the interface normal, and $\boldsymbol{\nabla}_A\sigma = \boldsymbol{\nabla}\sigma - \mathbf{n}(\mathbf{n}\cdot\boldsymbol{\nabla}\sigma)$ is the surface gradient of the variable surface tension coefficient. This surface term, tangential to the interface, is called *Marangoni stress*. For a constant σ, this term is obviously zero. By integrating $\mathbf{f}_\sigma$ over the surface A [49]:

$$\int_A \mathbf{f}_\sigma \, dA = \int_V \mathbf{f}_\sigma \delta_A(\mathbf{x}-\mathbf{x}_A)\, dV \tag{1.14}$$

where $\delta_A(\mathbf{x}-\mathbf{x}_A)$ is a surface Dirac δ-function, which is non-zero only at the interface location $\mathbf{x}=\mathbf{x}_A$. As a consequence, the surface tension force per unit volume $\mathbf{F}_\sigma$ is:

$$\mathbf{F}_\sigma = \mathbf{f}_\sigma\delta_A(\mathbf{x}-\mathbf{x}_A) = (\sigma\kappa\mathbf{n} + \boldsymbol{\nabla}_A\sigma)\delta_A(\mathbf{x}-\mathbf{x}_A) \tag{1.15}$$

By defining $\delta_A = \delta_A(\mathbf{x}-\mathbf{x}_A)$ and using a constant surface tension coefficient, which will be the case in this work, the volumetric surface tension force reduces to:

$$\boxed{\mathbf{F}_\sigma = \sigma\kappa\mathbf{n}\delta_A} \tag{1.16}$$

Thus, the surface tension volumetric force is proportional to the constant surface tension coefficient σ and to the curvature κ, is normal to the interface, and takes place only at the interface location as a singular distribution [51].

As a starting point for volumetric numerical models of surface tension, the surface Dirac δ-function $\mathbf{n}\delta_A$ is written as the gradient of a Heaviside function H, which yields the following reformulated analytical expression:

$$\mathbf{F}_\sigma = \sigma\kappa\boldsymbol{\nabla}H \tag{1.17}$$

where H is equal to 1 in liquid, and to 0 in gas. The central stake in volumetric numerical models of surface tension is to find a proper approximation for the sharp Heaviside function H [50]. This point will be discussed later in this chapter (Section 1.4).

1.1.3.2.2 Integral formulation

In two dimensions, using a constant surface tension coefficient and the curvilinear abscissa s to parametrize the portion of interface A in the control volume V [50]:

$$\int_V \mathbf{F}_\sigma \, dV = \int_V \sigma\kappa\mathbf{n}\delta_A \, dV = \int_A \sigma\kappa\mathbf{n} \, ds \tag{1.18}$$

Using the first Frenet–Serret formula, i.e. $\kappa\mathbf{n}ds = \mathbf{dt}$ with $\mathbf{dt}$ the infinitesimal tangent vector to the interface, the previous integral expression yields:

$$\int_A \sigma\kappa\mathbf{n} \, ds = \int_A \sigma \, \mathbf{dt} = \sigma \int_A \mathbf{dt} \tag{1.19}$$

which eventually writes:

$$\boxed{\int_V \mathbf{F}_\sigma \, dV = \sigma\mathbf{t}} \tag{1.20}$$

with $\mathbf{t}$ the unit tangent vector to the interface. In the case of a closed interface contour, the net force exerted by surface tension is zero:

$$\oint_A \sigma \, \mathbf{dt} = \sigma \oint_A \mathbf{dt} = \mathbf{0} \tag{1.21}$$

Thus, from a mechanical point of view, surface tension itself is a contact force per unit length tangential to the interface that has no effect on the variation of total momentum, which is probably the most intuitive way to represent it: one easily visualizes a curve under tension. However, its volumetric resultant is a body force normal to the interface: it represents a force resisting the extension of the surface [48, 51, 52].

1.1.3.3 Formulation with jump conditions

Using this formulation, classical incompressible Navier-Stokes equations Eqs. (1.1, 1.2) are written for each of the adjacent phases separately, and jump conditions are used for the sharp interface boundary conditions between the two fluids.

Without phase change and with incompressible viscous fluids, the normal kinematic condition at a liquid-gas interface Γ, derived from the mass flow rate conservation across the interface, yields:

$$\mathbf{u}_{l,\Gamma} \cdot \mathbf{n} = \mathbf{u}_{g,\Gamma} \cdot \mathbf{n} \tag{1.22}$$

where $\mathbf{u}_{l,\Gamma}$ and $\mathbf{u}_{g,\Gamma}$ are liquid and gas velocities at the interface, respectively. If we assume that on the interface the tangential velocities are continuous [53], the kinematic condition writes:

$$\boxed{[\mathbf{u}]_\Gamma = \mathbf{u}_{l,\Gamma} - \mathbf{u}_{g,\Gamma} = \mathbf{0}} \tag{1.23}$$

with $[.]_\Gamma$ the jump across the interface. The kinematic condition thus means that the velocity field $\mathbf{u}$ is continuous across Γ.

For an incompressible Newtonian fluid, using Eq. (1.10), the total stress tensor is:

$$\mathbf{T} = -p\mathbf{I} + \boldsymbol{\tau} = \underbrace{-p\mathbf{I}}_{\text{pressure stresses}} + \underbrace{\mu\left(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T\right)}_{\text{viscous stresses}} \tag{1.24}$$

The balance of stresses on a unit area A of interface Γ gives [54]:

$$\mathbf{f}_\sigma + [\mathbf{T}]_\Gamma \cdot \mathbf{n} = \mathbf{0} \tag{1.25}$$

with $\mathbf{f}_\sigma$ the surface tension stress given by Eq. (1.13):

$$\sigma\kappa\mathbf{n} + \boldsymbol{\nabla}_A\sigma + \left[-p\mathbf{I} + \mu\left(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T\right)\right]_\Gamma \cdot \mathbf{n} = \mathbf{0} \tag{1.26}$$

Considering only normal stresses, the previous equation yields:

$$\sigma\kappa + \mathbf{n}^T \cdot \left[-p\mathbf{I} + \mu\left(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T\right)\right]_\Gamma \cdot \mathbf{n} = 0 \tag{1.27}$$

Hence:

$$\sigma\kappa - [p]_\Gamma + \mathbf{n}^T \cdot \left[\mu\left(\boldsymbol{\nabla}\mathbf{u} + (\boldsymbol{\nabla}\mathbf{u})^T\right)\right]_\Gamma \cdot \mathbf{n} = 0 \tag{1.28}$$

Finally, the normal momentum jump condition for Newtonian incompressible fluids is:

$$\boxed{[p]_\Gamma = p_{l,\Gamma} - p_{g,\Gamma} = \sigma\kappa + 2\,[\mu]_\Gamma\,\mathbf{n}^T \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{n}} \tag{1.29}$$

where $p_{l,\Gamma}$ and $p_{g,\Gamma}$ are liquid and gas pressures at the interface, respectively. With this formulation, the pressure jump and thus surface tension forces can be embedded in the pressure gradient term of the momentum equation Eq. (1.2). This approach, formulated using jump conditions, will be used in this work. In the case of static fluids, Eq. (1.29) becomes:

$$[p]_\Gamma = p_{l,\Gamma} - p_{g,\Gamma} = \sigma\kappa \tag{1.30}$$

which is the well-known Young–Laplace equation, relating pressure and surface tension for an interface in equilibrium [55, 56].

1.1.4 Non-dimensional numbers

This section provides a list of the non-dimensional numbers that will be used in this work.

- The Reynolds number is the ratio between inertia and diffusion:

$$\mathrm{Re} = \frac{\rho|\mathbf{u}|L_c}{\mu} = \frac{|\mathbf{u}|L_c}{\nu} \sim \frac{\text{momentum convection}}{\text{momentum diffusion}} \tag{1.31}$$

  with L_c a characteristic length scale, and $\nu = \mu/\rho$ the kinematic viscosity.

- The Weber number is the ratio between inertia ad surface tension:

$$\mathrm{We} = \frac{\rho|\mathbf{u}|^2 L_c}{\sigma} \sim \frac{\text{inertial forces}}{\text{cohesive forces}} \tag{1.32}$$

- The Laplace number can be seen as a Reynolds number based on the visco-capillary velocity σ/μ:

$$\mathrm{La} = \frac{\mathrm{Re}^2}{\mathrm{We}} = \frac{\rho\sigma L_c}{\mu^2} \sim \frac{\text{cohesive forces}}{\text{viscous forces}} \tag{1.33}$$

- The Ohnesorge number, which is related to the Laplace number by an inverse relationship:

$$\mathrm{Oh} = \frac{1}{\sqrt{\mathrm{La}}} = \frac{\sqrt{\mathrm{We}}}{\mathrm{Re}} = \frac{\mu}{\sqrt{\rho\sigma L_c}} \sim \frac{\text{viscous forces}}{\text{cohesive forces}} \tag{1.34}$$

- The Capillary number is the flow velocity made non-dimensional using the visco-capillary velocity:

$$\mathrm{Ca} = \frac{|\mathbf{u}|}{\sigma/\mu} = \frac{|\mathbf{u}|\mu}{\sigma} \sim \frac{\text{flow velocity}}{\text{visco-capillary velocity}} \tag{1.35}$$

1.2 Numerical resolution of incompressible Navier-Stokes equations

This section aims at introducing the numerical framework used in this thesis and its stakes to solve the incompressible Navier-Stokes equations in the general case using the YALES2 flow solver [38]. Specific features involved for simulating interfacial incompressible two-phase flows will be presented in the subsequent chapter.

1.2.1 Projection method

The projection method, widely used for simulating incompressible flows, is based on Helmholtz' theorem. This fundamental theorem of vectorial calculus allows to decompose any smooth vectorial field into an irrotational field (curl free) and a solenoidal field (divergence free). Applied to the flow velocity field, it writes:

$$\mathbf{u} = \mathbf{u}_i + \mathbf{u}_s \tag{1.36}$$

with $\mathbf{u}_i$ and $\mathbf{u}_s$ the irrotational and solenoidal velocity fields. The following two relations are thus verified:

$$\begin{cases} \boldsymbol{\nabla} \times \mathbf{u}_i = \mathbf{0} \\ \boldsymbol{\nabla} \cdot \mathbf{u}_s = 0 \end{cases} \tag{1.37}$$

The irrotational part $\mathbf{u}_i$ is a conservative vector field, which means that it derives from a scalar potential φ:

$$\mathbf{u}_i = \boldsymbol{\nabla}\varphi \tag{1.38}$$

Applying the divergence operator to Eq. (1.36) yields:

$$\boldsymbol{\nabla} \cdot \mathbf{u} = \boldsymbol{\nabla} \cdot \mathbf{u}_i + \underbrace{\boldsymbol{\nabla} \cdot \mathbf{u}_s}_{=0} = \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla}\varphi) \tag{1.39}$$

eventually:

$$\boxed{\boldsymbol{\nabla} \cdot \mathbf{u} = \boldsymbol{\nabla}^2\varphi} \tag{1.40}$$

which is a Poisson equation for φ. This type of equation is classified as an elliptic partial differential equation (PDE), and its resolution allows to determine φ from $\mathbf{u}$, to then find the irrotational part of the velocity field $\mathbf{u}_i$ with Eq. (1.38). Enventually, the solenoidal part is $\mathbf{u}_s = \mathbf{u} - \mathbf{u}_i$, which allows to enforce the solenoidal nature of a velocity field: it is hence well-suited for the numerical resolution of incompressible flows. The velocity field is then said to be *projected*.

The application of this method to the incompressible Navier-Stokes equations yields to three steps, to advance the velocity field in time from $\mathbf{u}^n$ at time t^n to $\mathbf{u}^{n+1}$ at time t^{n+1}. These steps have been formulated by Chorin (1967) [57] as follows:

(i) To obtain a first approximation of the velocity field $\mathbf{u}^{n+1}$ from $\mathbf{u}^n$, one writes the momentum conservation equation without the pressure gradient, which only contributes to the irrotational part of the velocity field $\mathbf{u}_i$. One then formulates an equation to explicitly compute a *predicted velocity* $\mathbf{u}^*$ from $\mathbf{u}^n$:

$$\boxed{\frac{\rho\mathbf{u}^* - \rho\mathbf{u}^n}{\Delta t} = \boldsymbol{\nabla}\cdot\Big(\mu\left(\boldsymbol{\nabla}\mathbf{u}^n + (\boldsymbol{\nabla}\mathbf{u}^n)^T\right)\Big) - \boldsymbol{\nabla}\cdot\Big(\rho\mathbf{u}^n \otimes \mathbf{u}^n\Big)} \tag{1.41}$$

Please note that this prediction equation is written in conservative form, consistently with what will be presented subsequently in this manuscript.

(ii) The pressure gradient is taken into account as:

$$\frac{\mathbf{u}^{n+1}-\mathbf{u}^{*}}{\Delta t}=-\frac{1}{\rho}\boldsymbol{\nabla}p^{n+1/2} \tag{1.42}$$

which is then impossible because $p^{n+1/2}$ is unknown at this point. However, by applying the divergence operator to Eq. (1.42), one gets:

$$\frac{\boldsymbol{\nabla}\cdot\mathbf{u}^{n+1}-\boldsymbol{\nabla}\cdot\mathbf{u}^{*}}{\Delta t}=\boldsymbol{\nabla}\cdot\left(-\frac{1}{\rho}\boldsymbol{\nabla}p^{n+1/2}\right) \tag{1.43}$$

Then, enforcing $\boldsymbol{\nabla}\cdot\mathbf{u}^{n+1}=0$ yields:

$$\boxed{\frac{1}{\Delta t}\boldsymbol{\nabla}\cdot\mathbf{u}^{*}=\boldsymbol{\nabla}\cdot\left(\frac{1}{\rho}\boldsymbol{\nabla}p^{n+1/2}\right)} \tag{1.44}$$

which is a *Poisson equation* for pressure.

(iii) Once $p^{n+1/2}$ is known, its gradient (i.e. the irrotational part of the predicted velocity field $\mathbf{u}_i^*$) can be computed and soustracted to the predicted velocity field, leading to the solenoidal updated velocity field $\mathbf{u}_s^*=\mathbf{u}^{n+1}$:

$$\boxed{\mathbf{u}^{n+1}=\mathbf{u}^{*}-\frac{\Delta t}{\rho}\boldsymbol{\nabla}p^{n+1/2}} \tag{1.45}$$

This step is referred as the *correction step*.

The application of this method in YALES2 is different. Indeed, Eq. (1.41) without the pressure term can lead to unphysical results due to the fractioning of time marching. The original method is thus modified in YALES2 as follows [58]:

(i) A first velocity predition $\hat{\mathbf{u}}$ is performed by considering the complete momentum equation, including the pressure field at time step n:

$$\boxed{\frac{\rho\hat{\mathbf{u}}-\rho\mathbf{u}^{n}}{\Delta t}=\boldsymbol{\nabla}\cdot\Big(\mu\left(\boldsymbol{\nabla}\mathbf{u}^{n}+(\boldsymbol{\nabla}\mathbf{u}^{n})^{T}\right)\Big)-\boldsymbol{\nabla}\cdot\Big(\rho\mathbf{u}^{n}\otimes\mathbf{u}^{n}\Big)-\boldsymbol{\nabla}p^{n-1/2}} \tag{1.46}$$

The similarities between the previous pressure field $p^{n-1/2}$ and the one to compute $p^{n+1/2}$ lead to an accurate velocity estimator $\hat{\mathbf{u}}$, which takes into account the boundary conditions applied to $\mathbf{u}^{n+1}$. The pressure term at $t^{n-1/2}$ is then soustracted to this estimator:

$$\boxed{\frac{\rho\mathbf{u}^{*}-\rho\hat{\mathbf{u}}}{\Delta t}=\boldsymbol{\nabla}p^{n-1/2}} \tag{1.47}$$

(ii) The updated pressure field $p^{n+1/2}$ is then obtained as in Chorin's projection method following the Poisson equation Eq. (1.44).

(iii) Eventually, the correction step is performed following Eq. (1.45).

The original momentum equation is indeed obtained by summing Eqs. (1.46), (1.47) and (1.42). Moreover, including the previous pressure gradient $\boldsymbol{\nabla}p^{n-1/2}$ at the prediction step leads to a reduction of the errors related to the time-stepping process. The linear solver used to solve the Poisson equation for pressure is the Deflated Preconditioned Conjugate Gradient of [59].

1.2.2 Turbulence modeling - Large-Eddy Simulation (LES)

Turbulence has been described by Richard Feynman as the most important unsolved problem in classical physics. This physical phenomenon is ubiquitous, and characterized by irregularities, vortical structures, and chaotic behaviors of the flow (see Fig. 1.2): it is indeed very sensitive to initial conditions ("butterfly effect"). Turbulence arises from flow instabilities, and covers a broad

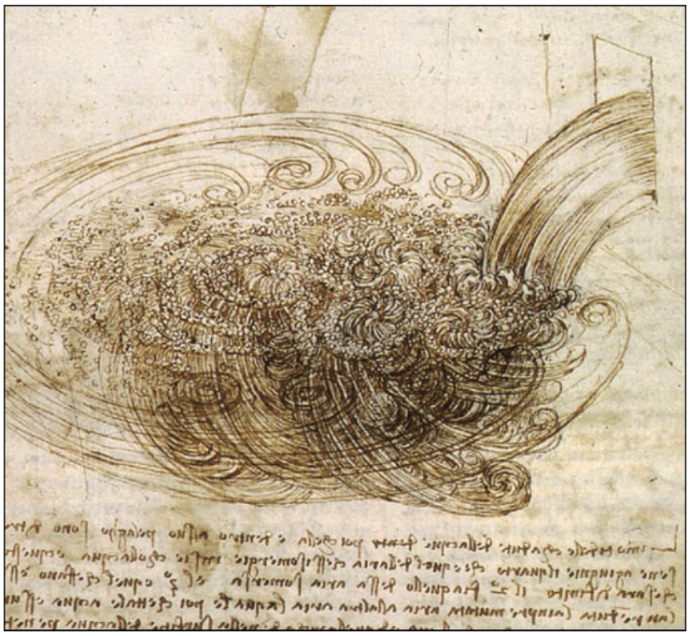


Figure 1.2: Fragment of *Studies of water passing obstacles and falling* by Leonardo da Vinci, c. 1508-1509 (Public Domain, from http://commons.wikipedia.org). This figure is reprinted from [60].

continuous range of spatial scales: from the largest scales l_e (*energetic* scales), to the smallest scales l_d (*dissipative* or *Kolmogorov* scales). As the smallest scales l_d are much greater than the molecular mean free path λ ($l_d \gg \lambda$; continuum hypothesis), the Navier-Stokes equations describe turbulence. The transition regime from laminar state to turbulent state can be characterized using the Reynolds number defined by Eq. (1.31). Turbulence is very important in industrial systems, as it enhances energy transfer.

There are three main approaches to simulate turbulent flows:

(i) *Direct Numerical Simulations (DNS)*: The Navier-Stokes equations, which intrinsically account for turbulence, are solved directly without any additional physical model. If one sets the cell size Δx to the same order of magnitude than the dissipative length scale l_d: $\Delta x \sim l_d$, all turbulent scales are resolved. Moreover, unsteady characteristics of the flow can be studied by employing DNS. However, the cost of these simulations can quickly become prohibitive, especially when atomization in realistic 3D industrial configurations is considered. Indeed, it has been demonstrated that the number of mesh points N_p required to compute all the turbulence scales in 3D are related to the Reynolds number based on the energetic length scale: $N_p \sim \mathrm{Re}_{l_e}^{9/4}$. DNS are thus suited for moderate Reynolds number flows [61];

(ii) *Reynolds-Averaged Navier-Stokes (RANS)* simulations: All scales of turbulence are modeled using statistical tools and a closure method through a turbulent viscosity μ_t. These closures

can be either algebraic (e.g. Goertler model, Prandtl mixing length), with one equation (e.g. Prandtl model, Spalart-Allmaras model), or with two equations (e.g. (k, ε) and (k, ω) models). Only the time-averaged flow characteristics are considered, but the restitution time is short: this approach is widely used in industry for this reason;

(iii) *Large-Eddy Simulations (LES)* is an intermediate approach between DNS and RANS. The largest turbulence scales of the flow are resolved, while vortical structures smaller than a cut-off length scale Δ are modeled from the large-scale informations, based on closure assumptions. This scale separation is performed by employing a spatial filtering operation. The cut-off scale Δ is usually closely related to the grid resolution, allowing to study the unsteady behavior of the flow. Large-Eddy Simulations allow to set the turbulence resolution cursor according to need or to computational resources available, leading to much cheaper simulations than when performing DNS, while still accessing unsteady characteristics of the flow. This is why it is starting to be used in industry.

In this work, DNS are performed for academic cases (Chapter 3), while LES is employed for 3D injection simulations (Chapter 4).

In the LES framework, a field ζ can be split up into a filtered part $\overline{\zeta}$ and a sub-filtered (or subgrid) field ζ':

$$\zeta(\mathbf{x}, t) = \overline{\zeta}(\mathbf{x}, t) + \zeta'(\mathbf{x}, t) \tag{1.48}$$

where the spatially-filtered field is defined by the following convolution:

$$\overline{\zeta}(\mathbf{x}, t) = \zeta * G_\Delta = \int_{\mathbb{R}^3} \zeta(\mathbf{x}', t) G_\Delta(\mathbf{x} - \mathbf{x}') \, \mathrm{d}\mathbf{x}' \tag{1.49}$$

with G_Δ the filter convolution kernel, associated with the cutoff length scale Δ. The scales $l < \Delta$ are filtered. If the wavenumber $k = 2\pi/l$ is introduced, the wavenumbers $k > k_\Delta = 2\pi/\Delta$ are filtered, leading to a low-pass filtering operation. The filter kernel G_Δ should satisfy the normalization condition:

$$\int_{\mathbb{R}^3} G_\Delta(\mathbf{x}) \, \mathrm{d}\mathbf{x} = 1 \tag{1.50}$$

and can commute, under conditions, with spatial and temporal derivatives:

$$\frac{\partial \zeta}{\partial \mathbf{x}} * G_\Delta = \frac{\partial}{\partial \mathbf{x}} (\zeta * G_\Delta) \quad \text{and} \quad \frac{\partial \zeta}{\partial t} * G_\Delta = \frac{\partial}{\partial t} (\zeta * G_\Delta) \tag{1.51}$$

As pointed out in [62], commutativity with spatial derivation operators is ensured if the filter G_Δ is isotropic, and if the filter size Δ is homogeneous. In practice, the spatial filtering is performed implicitly through the mesh, and hence the filter size Δ is closely related to the grid resolution. The commutation error is proportional to $d\Delta(x)/dx$, and can thus become non-negligible if local cell-size gradients are important. In our work, even with the use of Adaptive Mesh Refinement, cell-size gradients are smooth and controlled in the remeshing procedure, and thus, as it is usually done in LES, the spatial commutation is assumed. Furthermore, temporal commutation is ensured if the mesh does not evolve with time. If a moving mesh is employed (e.g. for simulating piston engines), which is not the case in this thesis, commutation errors need to be considered [63].

Using the properties detailed above, the Navier-Stokes equations can now be filtered. Let us consider the momentum equation, using the Einstein notation for convenience, i.e. writing the vectors as $\boldsymbol{\zeta} = \sum_{i=1}^{3} \zeta_i \mathbf{e}_i = \zeta_i \mathbf{e}_i$ and tensors as $\boldsymbol{\zeta} = \zeta_{i,j} \mathbf{e}_{i,j}$:

$$\frac{\partial(\rho \mathrm{u}_i)}{\partial t} + \frac{\partial(\rho \mathrm{u}_i \mathrm{u}_j)}{\partial x_j} = -\frac{\partial p}{\partial x_i} + \frac{\partial \tau_{i,j}}{\partial x_j} + \rho g_i \tag{1.52}$$

By removing gravity for convenience and filtering, the equation yields:

$$\overline{\frac{\partial(\rho \mathrm{u}_i)}{\partial t}} + \overline{\frac{\partial(\rho \mathrm{u}_i \mathrm{u}_j)}{\partial x_j}} = -\overline{\frac{\partial p}{\partial x_i}} + \overline{\frac{\partial \tau_{i,j}}{\partial x_j}} \tag{1.53}$$

Then, using a constant density together with spatial and temporal commutativity (Eq. (1.51)):

$$\frac{\partial(\rho \overline{\mathrm{u}_i})}{\partial t} + \frac{\partial(\rho \overline{\mathrm{u}_i \mathrm{u}_j})}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \overline{\tau_{i,j}}}{\partial x_j} \tag{1.54}$$

Using the relation $\overline{\mathrm{u}_i \mathrm{u}_j} = \bar{\mathrm{u}}_i \bar{\mathrm{u}}_j + \overline{\mathrm{u}'_i \mathrm{u}'_j} = \bar{\mathrm{u}}_i \bar{\mathrm{u}}_j + (\overline{\mathrm{u}_i \mathrm{u}_j} - \bar{\mathrm{u}}_i \bar{\mathrm{u}}_j)$, the filtered momentum equation becomes:

$$\frac{\partial(\rho \overline{\mathrm{u}_i})}{\partial t} + \frac{\partial(\rho \bar{\mathrm{u}}_i \bar{\mathrm{u}}_j)}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \overline{\tau_{i,j}}}{\partial x_j} + \frac{\partial}{\partial x_j}\Big(\underbrace{-\rho\left(\overline{\mathrm{u}_i \mathrm{u}_j} - \bar{\mathrm{u}}_i \bar{\mathrm{u}}_j\right)}_{\tau_{i,j}^{\mathrm{SGS}}}\Big) \tag{1.55}$$

The filtered momentum equation for incompressible flows eventually writes:

$$\boxed{\frac{\partial(\rho \overline{\mathrm{u}_i})}{\partial t} + \frac{\partial(\rho \bar{\mathrm{u}}_i \bar{\mathrm{u}}_j)}{\partial x_j} = -\frac{\partial \overline{p}}{\partial x_i} + \frac{\partial \overline{\tau_{i,j}}}{\partial x_j} + \frac{\partial \tau_{i,j}^{\mathrm{SGS}}}{\partial x_j}} \tag{1.56}$$

with $\tau_{i,j}^{\mathrm{SGS}}$ the subgrid Reynolds stress tensor. This term necessitates a closure: a subgrid model is hence needed.

The Boussinesq approximation [64] is the starting point of subgrid modeling. By analogy with the viscous stress tensor expression for incompressible Newtonian fluids Eqs. (1.3, 1.5, 1.10), the Boussinesq hypothesis writes:

$$\boxed{\tau_{i,j}^{\mathrm{SGS}} = 2\mu^{\mathrm{SGS}} \overline{S_{i,j}} = \mu^{\mathrm{SGS}} \left(\frac{\partial \overline{\mathrm{u}_i}}{\partial x_j} + \frac{\partial \overline{\mathrm{u}_j}}{\partial x_i}\right)} \tag{1.57}$$

with μ^{SGS} the subgrid viscosity (also called turbulent viscosity), and $\overline{S_{i,j}}$ the strain rate tensor of the filtered velocity field. Please note that the expression above only considers the deviatoric part of the subgrid scale tensor. The isotropic part $-(2/3)\rho \mathrm{k}^{\mathrm{SGS}} \delta_{i,j}$, which contains the turbulent kinetic energy $\mathrm{k}^{\mathrm{SGS}}$ not resolved by the grid, is usually neglected for low subgrid Mach numbers [65]. The subgrid viscosity μ^{SGS} is expressed using a subgrid model. To this aim, Smagorinsky [66] developed a model in which the turbulent viscosity is built as:

$$\boxed{\nu^{\mathrm{SGS}} = \frac{\mu^{\mathrm{SGS}}}{\rho} = (C_S \Delta)^2 \sqrt{2 \overline{S}_{i,j} \overline{S}_{i,j}}} \tag{1.58}$$

where C_S is the so-called Smagorinsky constant: $C_S \simeq 0.18$ in the case of homogeneous isotropic turbulence [67]. However, this model is often found to induce excessive damping of the resolved structures for transitional flows [68, 69]. To overcome this drawback, a dynamic estimation of the constant $C_S = C_S(\mathbf{x}, t)$ can be performed, to adjust locally in space and time to the turbulent flow. This idea led to the well-known dynamic Smagorinsky subgrid model [70, 71], which is based on the Germano identity [72, 73]. This identity allows any subgrid scale tensor computed at two different filtering levels Δ and Δ' (test filter) with $\Delta' > \Delta$ (usually $\Delta' = 2\Delta$), to be expressed analytically:

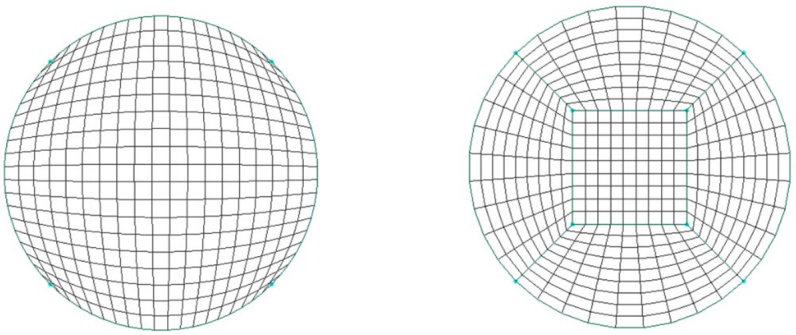
$$\mathcal{L}_{i,j} = \mathcal{T}_{i,j} - \widetilde{\tau_{i,j}^{\mathrm{SGS}}} \tag{1.59}$$

where $\widetilde{\tau_{i,j}^{\mathrm{SGS}}}$ is the subgrid scale tensor filtered by applying the test filter, $\mathcal{T}_{i,j} = \rho\left(\tilde{\bar{\mathrm{u}}}_i\tilde{\bar{\mathrm{u}}}_j - \widetilde{\bar{\mathrm{u}}_i\bar{\mathrm{u}}_j}\right)$ is the subgrid-scale tensor obtained by applying the test filter to the momentum equation Eq. (1.56), and $\mathcal{L}_{i,j} = \rho\left(\tilde{\bar{\mathrm{u}}}_i\tilde{\bar{\mathrm{u}}}_j - \widetilde{\bar{\mathrm{u}}_i\bar{\mathrm{u}}_j}\right)$ is the resolved turbulent stress calculated explicitly for the resolved velocity $\bar{\mathrm{u}}_i$. The underlying assumption is that a scale invariance exists in the wavenumber band defined by these two filters. The dynamic Smagorinsky model is used in the Large-Eddy Simulations performed in this thesis.

1.2.3 Meshing

In order to numerically solve the Navier-Stokes equations, the domain is partitioned into a set of elements, which entirely cover the domain and do not overlap [74].

In one dimension, the only way is to divide space into intervals. With more dimensions, there are more possibilities: the meshes can be either structured (or cartesian) or unstructured. For cartesian grids, which are easy to implement in computers, each neighbor may be found with a trivial numbering of the cells: (i, j) in 2D or (i, j, k) in 3D. The grid lines are not necessarly parallel to each other, they may be curved for instance. These meshes are more suitable for simple geometries: it is for example impossible to tesselate a disk with a structured grid. They may also be costly if fine resolution is locally required. Thus, a popular extension technique for cartesian meshes is to use multi-block structured grids: the domain is partitioned into several structured blocks, each of them containing a certain number of elements. With this framework, a numbering (i, j) or (i, j, k) is not mandatory, and meshing of curved geometries becomes possible: e.g. H-grid, O-grid, see Fig. 1.3 (a) and (b).

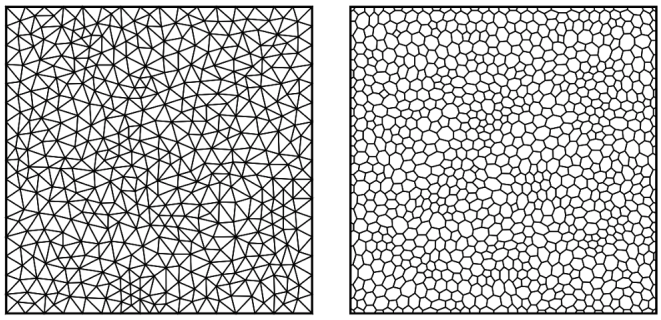


(a) H-grid (b) O-grid

Figure 1.3: Examples of multi-block structured meshes.

The goal in this thesis is to design algorithms for simulations in very complex geometries, i.e. applicable to realistic industrial systems, including sharp edges, narrow corners. To handle these complex geometries, the most suitable body-fitted mesh topology is unstructured grids: they can indeed fit an arbitrary domain boundary [74]. Less cells are usually required with this type of meshes, but are way more complicated to implement: each cell needs to have an identificator and the neighbors are found with a given connectivity, which leads a priori to less efficiency compared to cartesian grids, where the cells can be accessed straighforwardly. In practice, unstructured grids are usually made of triangles, quadrilaterals, or arbitrary polygons in 2D (see Fig. 1.4), and tetrahedra, hexahedra or arbitrary polyhedra in 3D. In this work, simplicial linear meshes (i.e. triangular and tetrahedral) are used. Their generation is based on Delaunay triangulation: no point is inside the circumcircle of any triangle, which guarantees a good mesh quality.

In order to save computational resources and tesselate the computational domain according to



(a) Triangular grid (b) Polygonal grid

Figure 1.4: Examples of 2D unstructured meshes. Reprinted from [11].

need, Adaptive Mesh Refinement (AMR) techniques have arisen. A popular technique for solving incompressible two-phase flows on adaptive cartesian grids is tree-based AMR (quad/octree), as in the work of Popinet [20, 21]. An example of two-phase flow application of cartesian tree-based AMR using the Gerris flow solver is displayed in Fig. 1.5. These methods can also be applied for

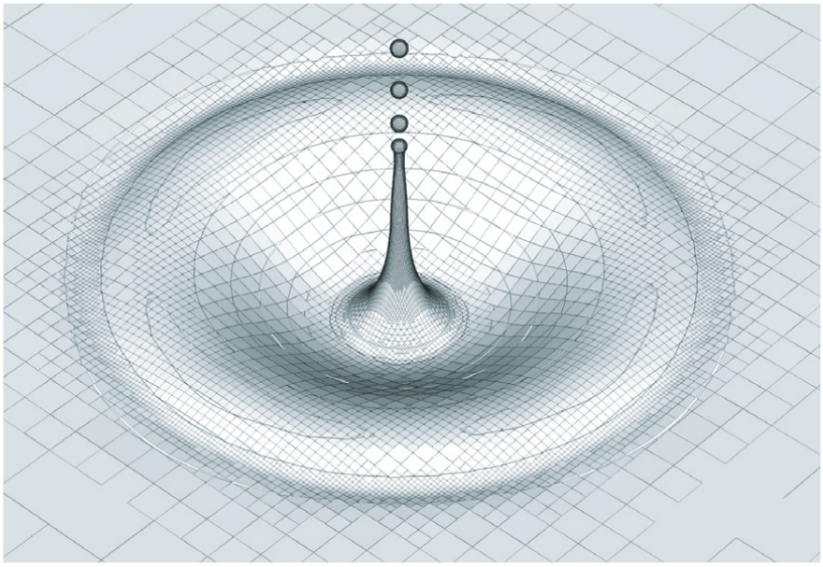


Figure 1.5: Jet created by the collapse under surface tension of an air bubble just below the surface of still water: the resolution varies along the interface according to local interface curvature. Reprinted from [75].

unstructured grids, but are way more complicated to implement [34].

In the present work, massively distributed, adaptive, and conforming (i.e. no hanging nodes) unstructured grids are employed. They are adapted using both metric-based and skewness-based quality criteria. For simplicial meshes (triangular and tetrahedral), the skewness $\mathcal{S}$ can be defined based on the equilateral volume as [76]:

$$\mathcal{S} = \frac{\text{optimal cell size} - \text{cell size}}{\text{optimal cell size}} \tag{1.60}$$

which means that $\mathcal{S} \to 1$ leads to a bad quality cell (e.g. a high aspect ratio triangle in 2D). The optimal cell size is the equilateral area (2D) or volume (3D): see Fig. 1.6 for an illustration

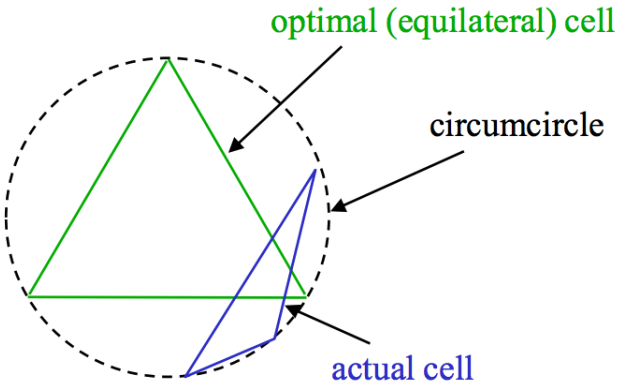


Figure 1.6: Illustration of the skewness computation based on the equilateral volume for a triangular mesh cell. Reprinted from [76].

with a triangular cell. If the skewness is too high, some solvers may have accuracy and stability issues. This method for determining the skewness is common and the default one for simplicial grids. Other methods exist, such as computing the skewness based on the deviation from a normalized equilateral angle, but are neither used nor detailed here: see Bakker (2008) [76] for more details. The external, sequential, remeshing library MMG [77] is used in this thesis, coupled to the YALES2 flow solver [78], so that parallel iterative remeshing-partitioning processes are performed dynamically around a moving material interface throughout a run. Such process can be done not only for interfacial two-phase flow applications, but also in the context of combustion. An example of such application is shown in Fig. 1.7, where a tetrahedral mesh is dynamically refined around the flame front in a semi-industrial burner. As seen in the figure, a fine resolution

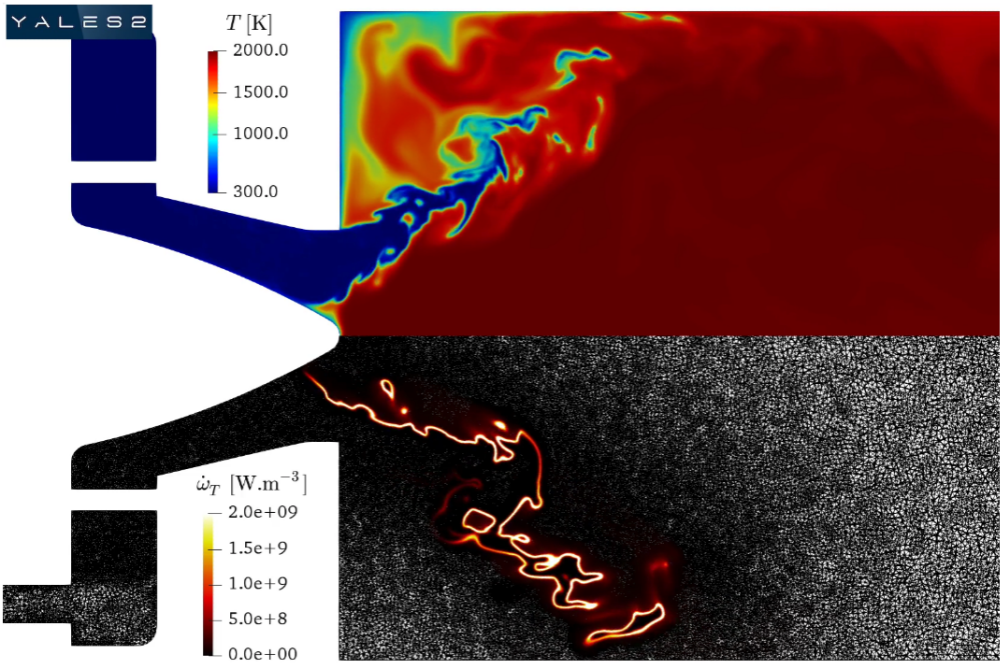


Figure 1.7: Dynamic mesh adaptation using YALES2/MMG for combustion applications [79, 80]. The top slice depicts the temperature field T, while the bottom one shows the heat release $\dot{\omega}_T$ together with the adapted tetrahedral grid.

is imposed around the flame front, and a smooth increase in cell size occurs as the distance to the interface increases: we will come back to this point later in the manuscript (Section 2.4.1). The YALES2/MMG parallel mesh adaptation strategy relies on a two-level grid decomposition: cells are gathered in cell groups, themselves gathered within a processor. The double-domain decomposition of YALES2 is presented and illustrated in Section 1.2.6 of the present chapter. The technical details about the two-level parallel dynamic grid adaptation technique, in the context of interfacial two-phase flows, are presented in the subsequent chapter (Section 2.4).

1.2.4 Finite-Volume discretization - Control volume

In the finite-volume framework for spatial discretization, the integral form of the conservation equations are used as a starting point. The computational domain is divided into a finite number of control volumes V, and the equations are integrated over each of these control volumes [74]. This framework allows to conserve well physical quantities inside the domain, as a flux leaving a control volume enters the neighboring one. The momentum equation in integral form writes:

$$\int_V \frac{\partial(\rho\mathbf{u})}{\partial t}\,dV + \int_V \boldsymbol{\nabla}\cdot(\rho\mathbf{u}\otimes\mathbf{u})\,dV = -\int_V \boldsymbol{\nabla}p\,dV + \int_V \boldsymbol{\nabla}\cdot\boldsymbol{\tau}\,dV + \int_V \rho\mathbf{g}\,dV \tag{1.61}$$

Using the divergence theorem, the equation becomes:

$$\int_V \frac{\partial(\rho\mathbf{u})}{\partial t}\,dV + \oint_A (\rho\mathbf{u}\otimes\mathbf{u})\cdot\mathbf{dA} = -\oint_A p\,\mathbf{dA} + \oint_A \boldsymbol{\tau}\cdot\mathbf{dA} + \int_V \rho\mathbf{g}\,dV \tag{1.62}$$

with $\mathbf{dA} = \overline{\mathbf{n}}dA$ the oriented surface vector. For sake of simplicity, the gravity term is removed and the equation can be re-organized as:

$$\int_V \frac{\partial(\rho\mathbf{u})}{\partial t}\,dV = \oint_A \Big(-(\rho\mathbf{u}\otimes\mathbf{u})\cdot\overline{\mathbf{n}} - p\overline{\mathbf{n}} + \boldsymbol{\tau}\cdot\overline{\mathbf{n}}\Big)\,dA \tag{1.63}$$

For a classical cartesian *staggered* MAC grid, originally introduced for free-surface flow simulations (i.e. one fluid involved) [49] within the Marker-And-Cell (MAC) method of Harlow and Welch (1965) [81], the control volumes for pressure and velocity are shifted: pressure is stored at mesh cell centers, while velocity is defined at cell faces. Both are defined at the center of their respective control volumes. Thus, the control volumes for pressure coincide with the grid cells (i.e. based on the *primal mesh*), see Fig. 1.8 for notations.

However in the YALES2 incompressible flow solver [38] used in the present work, control volumes do not coincide with the grid, they are indeed based on the *dual mesh*, constructed from the centers of edges and simplicial elements (dual cells connect centroids of cells to centroids of faces). The flow data such as pressure and velocity are stored at grid nodes: they are *collocated*. The unstructured finite-volume schemes employed in this thesis are pair-based: index i will represent the node around which the control volume is defined, and k will represent a neighbor of the node i, in the set of neighbors $\mathcal{N}_i$. See Fig. 1.9 for notations. As it can be seen in the figure, the node-based control volumes are polyhedral. Eq. (1.63) is thus applied to each polyhedron of the dual mesh.

In YALES2, 4th-order central finite-volume schemes are written on the dual grid [38]. The average of a quantity ζ over the control volume V is approximated by its nodal value:

$$\boxed{\int_V \zeta\,dV \simeq \zeta_i V_i} \tag{1.64}$$

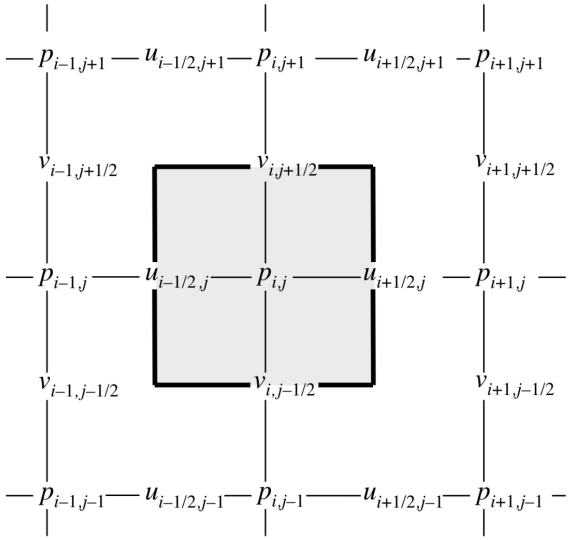


Figure 1.8: Notations for a staggered MAC grid. Reprinted from [49].

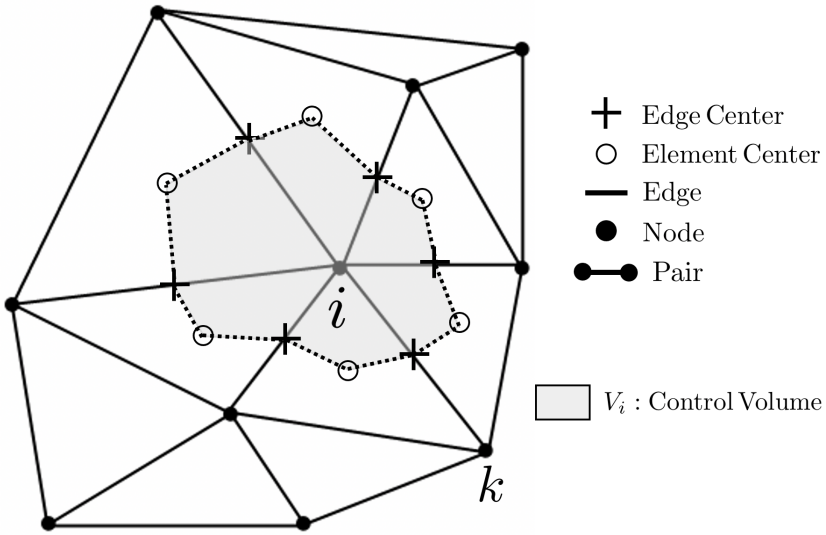


Figure 1.9: Grid notations for the YALES2 flow solver; Node-centered control volume.

As mentioned in [82], this expression is only first-order accurate for a generic mesh. It becomes second-order accurate if the control volume center of mass is located at the node position. This is only true if the mesh is locally uniform and isotropic. The discretization errors are greatly reduced by avoiding large cell-size ratios between adjacent cells. Thus, special care is required when refining the grid locally using isotropic Adaptive Mesh Refinement (AMR), so that a smooth decrease in grid size towards the region of interest is performed.
The integrated gradient operator applied to a scalar field ζ writes:

$$\int_V \boldsymbol{\nabla}\zeta \, dV = \oint_A \zeta \, \mathbf{dA} \simeq \boldsymbol{\nabla}\zeta|_i V_i \tag{1.65}$$

The integrated divergence operator applied to a flux $\mathbf{F}$ writes:

$$\int_V \boldsymbol{\nabla}\cdot\mathbf{F} \, dV = \oint_A \mathbf{F}\cdot\mathbf{dA} \simeq \boldsymbol{\nabla}\cdot\mathbf{F}|_i V_i \tag{1.66}$$

Eqs. (1.65) and (1.66) can now be used to obtain the discretized expressions for the node-based gradient and divergence operators. Second and fourth-order gradients of a quantity ζ are computed at a given node i using the gradient theorem as:

$$\boxed{\boldsymbol{\nabla}\zeta|_i = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\zeta_{ik}\mathbf{dA}_{ik}} \tag{1.67}$$

with $\mathcal{N}_i$ the set of neighbors of node i, ζ_{ik} the pair-based estimation of ζ, computed to either second or fourth order, and $\mathbf{dA}_{ik}$ the normal vector to the portion of the control volume V_i, associated to pair ik (red solid line in Fig. 1.10). Second and fourth-order approximations of the gradient are written as:

$$\boldsymbol{\nabla}\zeta|_i^{\text{2nd}} = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\left(\frac{\zeta_i+\zeta_k}{2}\right)\mathbf{dA}_{ik} \tag{1.68}$$

$$\boldsymbol{\nabla}\zeta|_i^{\text{4th}} = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\left(\frac{\zeta_i+\zeta_k}{2}+\frac{\boldsymbol{\nabla}\zeta|_i^{\text{2nd}}-\boldsymbol{\nabla}\zeta|_k^{\text{2nd}}}{6}\cdot\boldsymbol{\Delta x}_{ik}\right)\mathbf{dA}_{ik} \tag{1.69}$$

where $\boldsymbol{\Delta x}_{ik}$ is the direction of the pair ik. It is useful to define the unit pair direction $\mathbf{e}_{ik} = \boldsymbol{\Delta x}_{ik}/\Delta x_{ik}$, which will be used in the subsequent chapters. It should be noted that this discretization scheme is fourth-order accurate on quasi-regular grids. More details about the high-order finite-volume framework used can be found in [83]. Using the divergence theorem, the divergence of a flux $\mathbf{F}$ writes:

$$\boxed{\boldsymbol{\nabla}\cdot\mathbf{F}|_i = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\mathbf{F}_{ik}\cdot\mathbf{dA}_{ik}} \tag{1.70}$$

with $\mathbf{F}_{ik}$ the pair-based flux computed as in the gradient to either second or fourth order. To ease

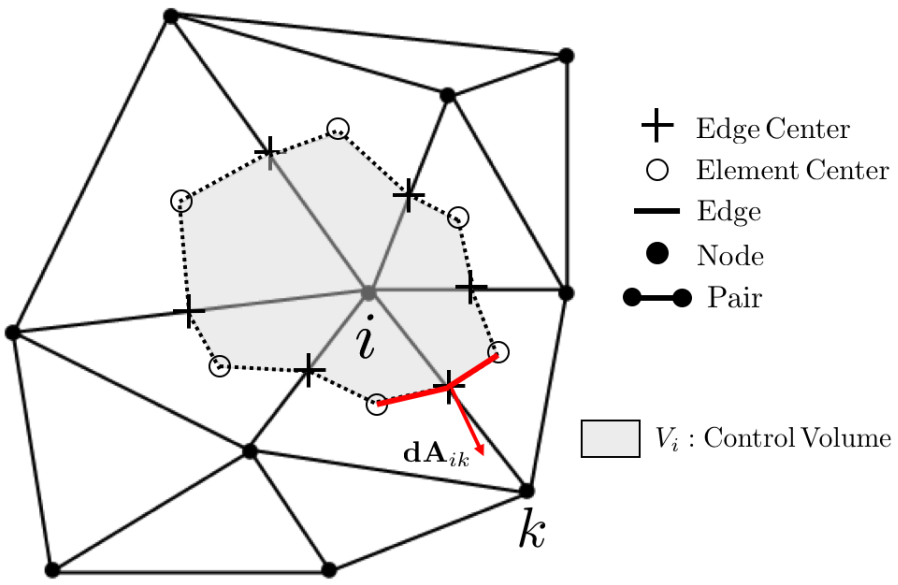


Figure 1.10: Finite-volume pair-based discretization in YALES2, using node-centered control volumes V_i.

the reading of the manuscript, the necessary details about the pair-based finite-volume schemes are recalled in the next chapter (Section 2.2.3).

1.2.5 Time integration

Explicit temporal integration is performed. Several schemes are available in YALES2, such as the fourth-order Runge-Kutta scheme (RK4). Let us consider the generic transport equation of a given quantity ζ:

$$\frac{\partial \zeta}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \zeta = 0 \tag{1.71}$$

where the flow velocity $\mathbf{u}$ is assumed constant. If one introduces $\mathcal{C}_i$ the discretized convection term at node i, a simple one-step discretized transport equation writes:

$$\zeta_i^{n+1} = \zeta_i^n - \Delta t \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) \tag{1.72}$$

The RK4 method decomposes the advancement into four steps as follows:

$$\begin{cases} \zeta_i^{(1)} = \zeta_i^n - \frac{\Delta t}{4} \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) \\ \zeta_i^{(2)} = \zeta_i^n - \frac{\Delta t}{3} \mathcal{C}_i \left(\zeta_i^{(1)}, \mathbf{u}_i\right) \\ \zeta_i^{(3)} = \zeta_i^n - \frac{\Delta t}{2} \mathcal{C}_i \left(\zeta_i^{(2)}, \mathbf{u}_i\right) \\ \zeta_i^{n+1} = \zeta_i^n - \Delta t \mathcal{C}_i \left(\zeta_i^{(3)}, \mathbf{u}_i\right) \end{cases} \tag{1.73}$$

When the fluxes are linearized with respect to ζ (constant convection velocity), the RK4 method is also called a Taylor method as it relies on a Taylor expansion of the Right-Hand Side (RHS).

The time-marching scheme used in this work, called TFV4A [84], builds upon the RK4 scheme, which can be rewritten as a two-step method by recursion as:

$$\begin{cases} \zeta_i^{(2)} = \zeta_i^n - \frac{\Delta t}{3} \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) + \frac{\Delta t^2}{12} \mathcal{C}_i^2 \left(\zeta_i^n, \mathbf{u}_i\right) \\ \zeta_i^{n+1} = \zeta_i^n - \Delta t \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) + \frac{\Delta t^2}{2} \mathcal{C}_i^2 \left(\zeta_i^{(2)}, \mathbf{u}_i\right) \end{cases} \tag{1.74}$$

where $\mathcal{C}^2 = \mathcal{C} \circ \mathcal{C} = \mathbf{u} \cdot \boldsymbol{\nabla}(\mathbf{u} \cdot \boldsymbol{\nabla} \zeta)$ is the twice-applied convection operator (4Δ-stencil). These two detailed expressions have been obtained by replacing $\zeta_i^{(1)}$ in the expression of $\zeta_i^{(2)}$, $\zeta_i^{(3)}$ in the expression of ζ_i^{n+1}, and then developing by linearity. The so-called TFV4A scheme used in this thesis is indeed a blend of the RK4 method Eq. (1.74) and a Lax-Wendroff type scheme, named TTG4A (Two-step Taylor-Galerkin). The TTG4A scheme is written as:

$$\begin{cases} \zeta_i^{(2)} = \zeta_i^n - \frac{\Delta t}{3} \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) + \frac{\Delta t^2}{12} \mathcal{D}_i \left(\zeta_i^n, \mathbf{u}_i\right) \\ \zeta_i^{n+1} = \zeta_i^n - \Delta t \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) + \frac{\Delta t^2}{2} \mathcal{D}_i \left(\zeta_i^{(2)}, \mathbf{u}_i\right) \end{cases} \tag{1.75}$$

where the twice-composed convection operator $\mathcal{C}^2$ in RK4 is replaced by a compact diffusion operator $\mathcal{D}$ (2Δ-stencil) in TTG4A. This is the only difference between the two schemes. A linear combination between RK4 and TTG4A yields:

$$\begin{cases} \zeta_i^{(2)} = \zeta_i^n - \alpha \frac{\Delta t}{3} \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) + (1-\alpha) \frac{\Delta t^2}{12} \mathcal{C}_i^2 \left(\zeta_i^n, \mathbf{u}_i\right) + \alpha \frac{\Delta t^2}{12} \mathcal{D}_i \left(\zeta_i^n, \mathbf{u}_i\right) \\ \zeta_i^{n+1} = \zeta_i^n - \alpha \Delta t \mathcal{C}_i \left(\zeta_i^n, \mathbf{u}_i\right) + (1-\alpha) \frac{\Delta t^2}{2} \mathcal{C}_i^2 \left(\zeta_i^{(2)}, \mathbf{u}_i\right) + \alpha \frac{\Delta t^2}{2} \mathcal{D}_i \left(\zeta_i^{(2)}, \mathbf{u}_i\right) \end{cases} \tag{1.76}$$

where α is an adjustable parameter, to set the impact of the diffusive terms. $\alpha = 1$ yields the TTG4A scheme which is stable and diffusive, while $\alpha = 0$ recovers the RK4 scheme.

1.2.6 Domain decomposition - Parallelization

High-Performance Computing (HPC) is an essential tool for simulating complex industrial processes. Indeed, for example, designing industrial injectors requires to simulate complex three-dimensional two-phase flows. Atomization especially, which leads to the formation of a large amount of droplets, involves a substantial number of degrees of freedom. Thus, large computational resources are necessary to run these simulations, and as a consequence CFD code performances by the mean of parallelization are at the core of the problem. To optimize a parallel CFD code, one has to ensure that the load on each processor is balanced. Indeed, at each global communication, processors will have to wait for each other, leading to a simulation roughly driven by the slowest processor. A special attention should then be dedicated to the initial mesh partitioning in order to distribute each piece of the computational domain to the various processors, in the most balanced way possible. In YALES2, to optimize this distribution and memory management, a double domain decomposition is performed.

In the double domain decomposition [38, 58], illustrated in Fig. 1.11, a first partition of the mesh is performed as in single domain decomposition methods. This operation leads to one cell group on each processor. Next, each processor will split its subdomain into cell groups, with the requested number of cells per group specified by the user at the beginning of the run. Between these groups, internal communicators are built, whose role is similar than the one of external communicators but which will not be concerned by the external parallelization library, which is the Message Passing Interface (MPI) library used in YALES2. This framework allows to form contiguous blocks in memory, so that data operations required for a computation on one group can be loaded simultaneously in the cache memory of the processor. This methods is then said "cache-aware", which greatly optimizes data transmissions between the processor's Random Access Memory (RAM) and its cache memory, leading to efficient data accesses. Moreover, one can adjust the group size to the cache memory capacity of the processors used for the run.

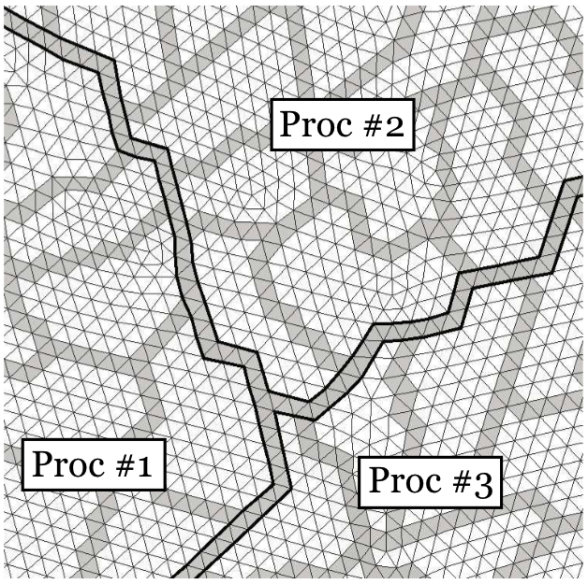


Figure 1.11: Double domain decomposition of a 2D triangular mesh. The cells located between black thick lines are those shared by different processors, and are thus involved in external communications (MPI). The grey cells are at the boundary of different cell groups within a processor (internal communications). Reprinted from [58].

The Deflated Preconditioned Conjugate Gradient (DPCG) linear solver of Malandain et al. [59] used in this work to solve the pressure Poisson equation in the context of the projection method (presented in Section 1.2.1) is optimized by using the so-called double domain decomposition. Indeed, the DPCG algorithm is similar to multigrid methods and hence requires to deal with computational operations on a coarse grid: the double domain decomposition is hence well-suited, convenient, to perform these operations. Eventually, local refinement of the mesh can be performed directly on cell groups, and thus internal and external communicators associated with the cell group do not necessitate to be rebuilt.

The global partitioning for the repartition of the cells on processors and the local partitioning into cell groups on each processor are performed using the METIS library [85].

1.3 Sharp interface-capturing techniques

For liquid-gas multiphase flows, in order to numerically describe the evolution of the liquid-gas interface dynamics and topology in time and space, as well as to compute the material properties and surface tension forces, dedicated numerical methods and models are needed. The methods available for such purpose are numerous, but can be gathered into several main categories of methods.

The first generation of techniques derives from the pioneering Marker-And-Cell (MAC) method (1965) [81], which was originally used for simulating free-surface flows (i.e. one fluid involved), but evolved later to compute multifluid problems. The MAC method consists in placing Lagrangian virtual marker particles in Eulerian cells and advecting them with the updated velocity field. If a cell contains a particle, it is considered to contain fluid. This way, flow visualization of the free surface is possible [86]. The idea of using Lagrangian markers to represent and advect an interface has led to another method, namely the Front-Tracking (FT) method (1992) [87, 88, 89]. In this method, the interface is a chain of markers connected by segments (called *front*), which yields an explicit description of the interface position (see Fig. 1.12). The advantage of the FT

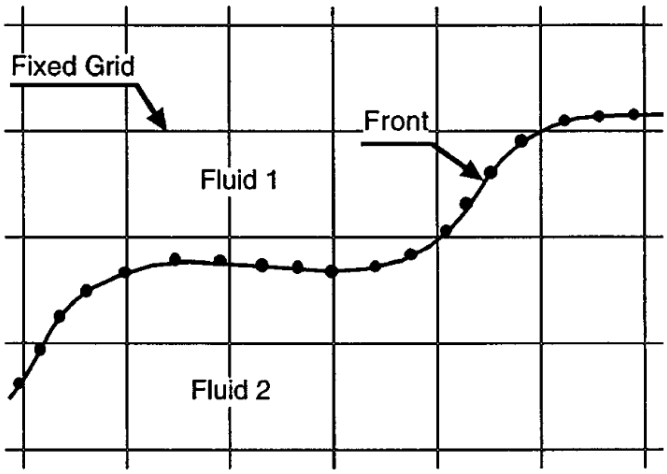


Figure 1.12: Illustration of the Front-Tracking method. Reprinted from [89].

method is its accuracy, as it allows sub-grid resolution of the interface dynamics. However, it does not implicitly predict topology changes, and thus an additional explicit criterion is required. The MAC and FT techniques are gathered in the *interface-tracking* class of methods (Lagrangian advection of the liquid-gas interface).

On the other hand, Eulerian interface advection techniques, classified into the *interface-capturing* class, are the most widely-used methods for simulating material interface dynamics. They rely on the transport of an implicit function, used to identify the phases and describe the interface geometry. The main interface-capturing methods are the Volume-Of-Fluid (VOF) [90] and the Level Set (LS) techniques [91, 92, 4]. The VOF method uses and advects the volume fraction to account for the interface presence, while in the classical LS method the interface is located as the zero-level of a regular, signed-distance function.

Besides properly simulating the interface dynamics, one needs to compute the material properties, such as density and viscosity, from the interface location. On one hand, for *sharp-interface* techniques, the interface is a discontinuity and thus the density and viscosity values encounter a jump across the interface: a Heaviside function can thus a priori be used, but is often regularized for numerical reasons. The VOF, LS, MAC, and FT methods belong to this sharp-interface category. On the other hand, in *diffuse-interface* methods, a physical interface thickness is defined and these material properties are continuous functions of the local phase indicator function. The diffuse-interface methods include in particular Phase-Field (PF) techniques [93, 94, 95], based on the convection-diffusion equation of the phase, by introducing physical effects that govern interfaces in a thermodynamically consistent way [96, 49]. Another technique in this category is the Second-Gradient (SG) method [97, 98, 99, 100], which considers, within the continuous transition zone, second-order density terms such as $(\boldsymbol{\nabla}\rho)^2$ in Navier-Stokes equations and in thermodynamic quantities. Eventually, one can mention the Constrained Interpolation Profile (CIP) method [101], in which the transition between the two fluids is described by a cubic polynomial [49].

The method used in this thesis to describe the liquid-gas interface dynamics belongs to interface-capturing techniques (Eulerian description), and derives from a sharp-interface framework (discontinuous material properties). As a consequence, an emphasis is put on this class of methods (i.e. VOF and LS methods) in the rest of this section. It should eventually be mentioned that the numerical methods for sharp interface capturing presented in detail in the following are classified according to Mirjalili et al. (2017) [102], and are introduced in the context of a finite-volume spatial discretization.

To ease the reading of the following sub-sections, a graphical classification of sharp interface-capturing methods is provided in Fig. 1.13.

1.3.1 Volume Of Fluid (VOF)

The VOF method was first introduced in [90] and is now one of the most famous and successful interface-capturing methods. The idea is to introduce a liquid volume fraction C, which is conserved throughout a simulation to numerically respect mass conservation. This function is the spatial average over a control volume V of a phase indicator function (Heaviside) H, defined as:

$$H(\mathbf{x},t) = \begin{cases} 1 \text{ if } \mathbf{x} \text{ is in liquid} \\ 0 \text{ if } \mathbf{x} \text{ is in gas} \end{cases} \tag{1.77}$$

Therefore, the volume fraction of liquid C, also known as the "color function", is:

$$C(\mathbf{x},t) = \frac{1}{V}\int_V H(\mathbf{x},t)\, dV \tag{1.78}$$

The phase indicator H is advected with the flow, its material derivative DH/Dt thus vanishes:

$$\frac{DH}{Dt} = \frac{\partial H}{\partial t} + \mathbf{u}\cdot\boldsymbol{\nabla}H = \frac{\partial H}{\partial t} + \boldsymbol{\nabla}\cdot(\mathbf{u}H) - H\boldsymbol{\nabla}\cdot\mathbf{u} = 0 \tag{1.79}$$

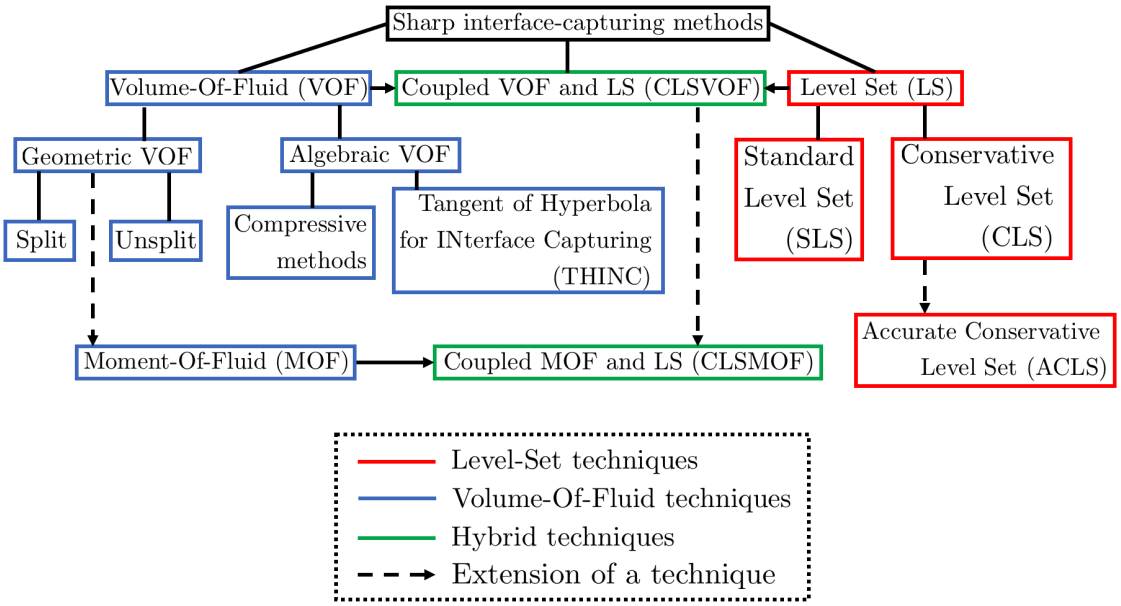


Figure 1.13: Classification of sharp interface-capturing techniques. Inspired by [102].

By integrating Eq. (1.79) over the control volume V:

$$\int_V \frac{\partial H}{\partial t}\, dV + \int_V \boldsymbol{\nabla} \cdot (\mathbf{u}H)\, dV - \int_V H \boldsymbol{\nabla} \cdot \mathbf{u}\, dV = 0 \tag{1.80}$$

Using Eq. (1.78) and the divergence theorem, Eq. (1.80) becomes:

$$V\frac{\partial C}{\partial t} + \int_A \mathbf{u}H \cdot \mathbf{dA} - \int_V H \boldsymbol{\nabla} \cdot \mathbf{u}\, dV = 0 \tag{1.81}$$

Even in the framework of incompressible flows (i.e. $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$), the third term may be kept for numerical implementation purposes [102]. Nonetheless, without this term, the semi-discrete advection equation of the color function C eventually yields:

$$\boxed{C^{n+1} = C^n - \frac{\Delta t}{V} \int_A \mathbf{F} \cdot \mathbf{dA}} \tag{1.82}$$

with $\Delta t = t^{n+1} - t^n$ the time step and $\mathbf{F} = \mathbf{u}H$ the flux across the control surface A. It should be noted that if Eq. (1.82) is summed all over the grid cells with correct boundary conditions, we get $\sum_{\text{cells}} C^{n+1} = \sum_{\text{cells}} C^n$, leading to liquid volume conservation over time [49].

The evaluation of the flux budget across the control surface $\int_A \mathbf{F} \cdot \mathbf{dA}$ requires a numerical approximation of the phase indicator H. This approximation can be either algebraic (e.g. a polynomial or trigonometric function) or geometric (e.g. via a line or plane equation, for 2D or 3D, respectively). These two types of approximation yield to two classes of VOF methods: algebraic and geometric VOF methods, respectively.

1.3.1.1 Algebraic VOF

Algebraic VOF methods are one of the first interface-capturing methods that have been developed. The original VOF method of Hirt and Nichols (1981) [90] belongs to this class. In algebraic VOF methods, the sharp phase indicator H is approximated numerically using algebraic functions

(volume-averaged, polynomial, hyperbolic tangent). Thus, algebraic VOF methods proceed in two steps: (i) Reconstruction of the fluxes $\mathbf{F}$ through an algebraic approximation of H; (ii) Advection of the volume-fraction field from C^n to C^{n+1}.

Depending on the numerical method chosen to compute the fluxes $\mathbf{F} = \mathbf{u}H$, algebraic VOF schemes can be divided into compressive schemes and THINC schemes, where THINC stands for Tangent of Hyperbola for INterface Capturing.

1.3.1.1.1 Compressive schemes

The idea of compressive schemes is to use the orientation of the interface normal with respect to the computational cell face normal [102], represented by the angle θ_f. A face flux $\mathbf{F}_f = \mathbf{u}_f C_f$, in which the color function C approximates the heaviside function H, is thus estimated. For each face f, upwind U, donor D, and acceptor A cells are defined, depending on the local flow direction given by the local face velocity field $\mathbf{u}_f$. An illustration of such methodology is provided in Fig. 1.14 for a polyhedral grid, reprinted from [48].

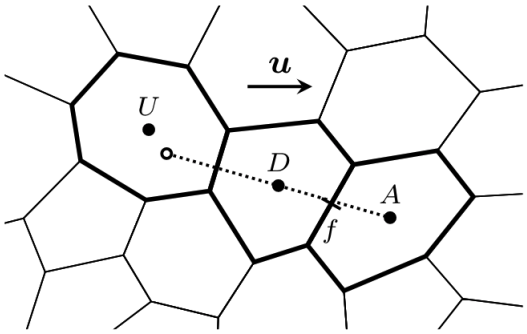


Figure 1.14: Illustration of upwind U, donor D, and acceptor A cells associated with a cell face f on a polyhedral mesh. U, D, and A are determined by the orientation of the local velocity field $\mathbf{u}_f$. Reprinted from [48].

In order to accurately estimate the face volume fraction C_f, a non-diffusive and bounded advection scheme is required. However, as mentioned in [103], the use of only a compressive scheme can cause an alignment of the fluid interface with the grid, whereas using only a diffusive scheme leads to a loss of accuracy when the flow is not oriented along a grid line. This issue gave birth to blended high-resolution (HR) schemes, mixing compressive and diffusive HR schemes. An HR scheme is a high-order scheme combined with the Convective Boundedness Criterion (CBC) [104], which ensures that neither undershoots nor overshoots will appear in the computational cell. Several examples of blending advection schemes exist, such as HRIC [105], CICSAM [106], STACS [107], HiRAC [108], and M-CICSAM [109].

The HR schemes can be formulated in the Normalized Variable Diagram (NVD) framework [110], which determines boundedness, stability and numerical diffusion conditions. In the NVD-CBC framework, a normalized face-based volume fraction $\tilde{C}_f$ is used:

$$\tilde{C}_f(\mathbf{x}) = \frac{C_f(\mathbf{x}) - C_\mathrm{U}}{C_\mathrm{A} - C_\mathrm{U}} \tag{1.83}$$

together with:

$$\tilde{C}_\mathrm{D}(\mathbf{x}) = \frac{C_\mathrm{D}(\mathbf{x}) - C_\mathrm{U}}{C_\mathrm{A} - C_\mathrm{U}} \tag{1.84}$$

where the subscripts A, U and D stand for acceptor, upwind, and donor cells, respectively, determined by the flow direction [109]. Thus, using this normalization process, $\tilde{C}_{\mathrm{U}} = 0$ and $\tilde{C}_{\mathrm{A}} = 1$. Thanks to the CBC, $\tilde{C}_f = f(\tilde{C}_{\mathrm{D}})$. The blending between a compressive downwind scheme and a diffusive high-resolution scheme, also called "switching strategy" is written as [109]:

$$\tilde{C}_{f,\mathrm{blend}} = \gamma_f(\theta_f)\tilde{C}_{f,\mathrm{compressive}} + (1 - \gamma_f(\theta_f))\,\tilde{C}_{f,\mathrm{diffusive}} \tag{1.85}$$

with $\gamma_f(\theta_f)$ the blending function ($0 \leq \gamma_f \leq 1$), depending on the interface orientation θ_f. As suggested by [108, 109], the blending function can be written as:

$$\gamma_f = \min\left(\left|\frac{\boldsymbol{\nabla} C|_{\mathrm{D}} \cdot \mathbf{d}}{|\boldsymbol{\nabla} C|_{\mathrm{D}}| \cdot |\mathbf{d}|}\right|^m, 1\right) \tag{1.86}$$

where $\mathbf{d}$ is the distance vector connecting the centers of the donor and acceptor cells (i.e. along a grid line), and $\boldsymbol{\nabla} C|_{\mathrm{D}}$ is the interface normal in the donor cell. The idea is to use the compressive scheme when the interface normal is aligned with the grid line, the diffusive scheme when the vectors are perpendicular, and a mix of the two in all intermediate situations.

However, the blended $\tilde{C}_{f,\mathrm{blend}}$ cannot be expressed in terms of nodal values at the center of the cells from Eq. (1.85). To overcome this difficulty, several implementation methods of blended HR schemes have arisen. Among these methods [111, 112, 113, 114], the Downwind Weighting Factor (DWF) method [112] can be cited as an example. It relies on the use of a face-based weighting factor $\beta_f(\tilde{C}_{f,\mathrm{blend}}, \theta_f)$, which carries all the information about the fluid distribution ($\tilde{C}_{f,\mathrm{blend}}$) and interface orientation (θ_f) so that:

$$\boxed{\mathbf{F}_f = \beta_f(\tilde{C}_{f,\mathrm{blend}}, \theta_f)\mathbf{F}_{f,\mathrm{A}} + \Big(1 - \beta_f(\tilde{C}_{f,\mathrm{blend}}, \theta_f)\Big)\,\mathbf{F}_{f,\mathrm{D}}} \tag{1.87}$$

with the downwind weighting factor β_f expressed as:

$$\beta_f = \frac{\tilde{C}_{f,\mathrm{blend}} - \tilde{C}_{\mathrm{D}}}{1 - \tilde{C}_{\mathrm{D}}} \tag{1.88}$$

As mentioned in [102], the accuracy of compressive algebraic VOF methods is generally an order of magnitude lower than recent geometric VOF methods, and depends on the local CFL criterion. Another drawback is the potential need to add an artificial interface sharpening term for flows with high strain rates, in order to avoid numerical diffusion.

1.3.1.1.2 THINC schemes

The idea of THINC schemes is to approximate the phase indicator function H by a hyperbolic tangent function $\tilde{H}$. The THINC method has been originally proposed in 1D in 2005 by [32], and revisited in 2011 in [33]. It relies on the use of the following piecewise interpolation function:

$$\boxed{\tilde{H}(x) = \frac{\alpha}{2}\left(1 + \gamma \tanh\left(\beta f(x, x_\Gamma)\right)\right)} \tag{1.89}$$

where α, γ and β are parameters which determine the quality of the solution. α ensures the boundedness of the function, γ is equal to 1 or -1 depending on the interface direction, and β sets the steepness of the profile:

$$\lim_{\beta \to \infty} \tilde{H}(x) = H(x) \tag{1.90}$$

The function $f(x, x_\Gamma)$ is a relative distance function to the position of the jump center x_Γ (i.e. the interface position). The original discretized expression for a 1D cartesian finite-volume scheme can be found in [32, 33].

The original extension of the THINC method to 2D and 3D cartesian grids, the MTHINC method (2012) [115] is done by using either a plane or a quadratic equation to represent the cell-wise interface. For a plane representation, the interface Γ is given by the equation:

$$n_x x + n_y y + n_z z + d = 0 \tag{1.91}$$

with $\mathbf{n} = (n_x, n_y, n_z)$ the interface normal, computed from the volume fraction C:

$$\mathbf{n} = \frac{\boldsymbol{\nabla} C}{|\boldsymbol{\nabla} C|} \tag{1.92}$$

and d is determined from the given volume fractions. The smoothed phase indicator thus writes:

$$\tilde{H}(\mathbf{x}) = \frac{1}{2}\left(1 + \tanh\left(\beta(n_x x + n_y y + n_z z + d)\right)\right) \tag{1.93}$$

The extension of this work to unstructured grids, the UMTHINC method, appeared in 2014 [116], with usage of a plane to represent the interface within a cell.

To improve the cell-wise representation of the interface in the UMTHINC method, a quadratic polynomial $\mathcal{P}(\mathbf{x})$ has been used in [117]:

$$\tilde{H}(\mathbf{x}) = \frac{1}{2}\left(1 + \tanh\left(\beta(\mathcal{P}(\mathbf{x}) + d)\right)\right) \tag{1.94}$$

This idea has been combined with a hybrid integration technique, blending 1D exact integration and Gaussian quadrature to integrate the flux $\mathbf{F} = \mathbf{u}\tilde{H}$ in Eq. (1.82). Indeed, integrating the multi-dimensional hyperbolic tangent function is another issue to tackle in the implementation of MTHINC/UMTHINC schemes. This blended spatial integration scheme then paves the way to a full Gaussian quadrature integration: the THINC/QQ scheme (THINC method with quadratic surface representation and Gaussian quadrature) of [118] (2017).

Recent studies report the use of a level set signed-distance function $\phi(\mathbf{x})$ to retrieve the quadratic polynomial $\mathcal{P}(\mathbf{x})$: the THINC/LS [119] (2018) and THINC/QQLS [120] (2019) methods, with Gaussian quadrature integration. The THINC scaling, including the level set function, writes:

$$\tilde{H}(\mathbf{x}) = \frac{1}{2}\left(1 + \tanh\left(\beta\phi(\mathbf{x})\right)\right) \tag{1.95}$$

where ϕ, the level set signed-distance function defined as $\phi(\mathbf{x}) = \pm|\mathbf{x} - \mathbf{x}_\Gamma|$, is approximated within a cell as:

$$\phi(\mathbf{x}) = \mathcal{P}(\mathbf{x}) \tag{1.96}$$

Thus, the interface is $\Gamma = \{\mathbf{x} \in \mathbb{R}^3 |\, \phi(\mathbf{x}) = \mathcal{P}(\mathbf{x}) = 0\}$. The usage of a level set function allows an accurate computation of geometrical informations (interface normal and curvature). However, as it will be seen later, the advection of the level set function ϕ does not ensure mass conservation, and thus a correction $\phi^\Delta(\mathbf{x})$ computed by a constraint on the volume fraction C is needed to enforce mass conservation [119]:

$$\tilde{H}(\mathbf{x}) = \frac{1}{2}\left(1 + \tanh\left(\beta(\mathcal{P}(\mathbf{x}) + \phi^\Delta(\mathbf{x}))\right)\right) \tag{1.97}$$

with $\phi^\Delta(\mathbf{x})$ computed so that:

$$\frac{1}{V}\int_V \tilde{H}\, dV = C^n \tag{1.98}$$

Hence, with this constraint, the interface is represented as $\Psi(\mathbf{x}) = \mathcal{P}(\mathbf{x}) + \phi^\Delta(\mathbf{x}) = 0$. This equation is named "Polynomial Surface of the Interface (PSI) equation" in [121] (2021).

1.3.1.2 Geometric VOF

A geometric VOF method requires two steps: (i) Explicit reconstruction of the interface from the cell-wise volume fractions C; (ii) Advection of the reconstructed interface in a given velocity field $\mathbf{u}$ using geometric methods, either direction by direction (split) or in one step (unsplit).

1.3.1.2.1 Interface reconstruction

The most widely used technique to explicitly reconstruct the interface within a computational cell is the Piecewise Linear Interface Calculation (PLIC) scheme of Debar (1974) [122]. This method consists in approximating the interface within a mesh cell as a line (2D) or a plane (3D). The equation of the cell-wise interface writes:

$$\boxed{\mathbf{n} \cdot \mathbf{x} + \alpha = n_x x + n_y y + n_z z + \alpha = 0} \tag{1.99}$$

where α is a constant, adjusted so that the liquid volume enclosed under the interface V_l equals $C \times V$, with V the total volume of the cell. The key part in PLIC reconstruction schemes is to find the local normal vector in cell k: $\mathbf{n}_k$.

In Youngs' method (1982) [123], the normal is computed from the normalized local gradient of volume fraction, discretized with finite differences:

$$\mathbf{n}_k = \frac{\boldsymbol{\nabla}_k C}{|\boldsymbol{\nabla}_k C|} \tag{1.100}$$

This method is fast and is accurate at low resolutions, i.e. when the radius of curvature is comparable to the cell size: second-order spatial convergence when approximating an ellipse [49]. However, the rate of convergence of the method lowers at high resolutions: first-order convergence for an ellipse reconstruction [49].

A method which shows better spatial convergence at high resolutions is the Centered-columns (CC) method of Scardovelli and Zaleski (2003) [124]. Presented in 2D for sake of simplicity, even if 3D extension for cartesian grids is possible, it relies on the use of height and width functions within a cell block, $y = f(x)$ and $x = g(y)$, constructed by summing the volumes fractions column-wise or row-wise, respectively. For a 3x3 block, one would compute the height functions y_{i-1} and y_{i+1}, and the width functions x_{j-1} and x_{j+1}. Then, the functions in the central column (y_i) and row (x_i) are estimated with a linear equation for each, where the slope, computed using a centered scheme, is the corresponding normal component value:

$$\begin{cases} \mathrm{sgn}(n_y) y = -n_x x + \alpha_1 \\ \mathrm{sgn}(n_x) x = -n_y y + \alpha_2 \end{cases} \tag{1.101}$$

where the signs of n_y and n_x are computed explicitly with centered finite differences, since $n_y = -\partial C/\partial y$ and $n_x = -\partial C/\partial x$. Indeed, summing the volume fractions yields to the loss of the information about the phases. However, depending on the interface topology and on how it cuts the borders of the cell block, only one of the two representations gives the correct slope. In some cases, the CC scheme may not even give the correct slope and an off-centered scheme is then necessary. To select the correct computed slope, n_x or n_y, the angular coefficient with the minimum absolute value is chosen. When several estimates are available for the same slope coefficient (e.g. $n_{x,1}$ and $n_{x,2}$), the coefficient with the minimum-absolute value is chosen. The CC method reconstructs well any linear interface thanks to the two additional criteria, but becomes weakly accurate when

interface curvature is important. This issue has led to the development of a two-segment cell-wise interface reconstruction by the same authors (2003) [124], to improve the representation of high-curvature regions.

To combine the good behaviors of Youngs' and Centered-columns methods at low and high resolutions, respectively, Aulisa et al. proposed the Mixed Youngs-Centered (MYC) method (2007) [125]. The best of the computed lines or planes with the CC method is first selected, then a selection between this best one and Youngs' method is done. The MYC method seems to be the most popular method for cartesian grids, as it shows a good accuracy/cost balance.

The Efficient Least-squares VOF Interface Reconstruction Algorithm (ELVIRA) technique, introduced by Pilliod and Puckett (2004) [126], also considers the height $y = f(x)$ and width $x = g(y)$ functions to compute an optimal linear/planar approximation of the cell-wise interface. In addition to the centered scheme to compute the slopes n_{xc} and n_{yc} as in the CC scheme, backward and forward estimates are also considered. This framework gives a total of 6 normal candidates in 2D ($\mathbf{n}_n$ with $n = 1...6$, from n_{xc}, n_{xf}, n_{xb}, n_{yc}, n_{yf}, and n_{yb}). First, the constant α_n is computed by enforcing volume conservation in the central cell. The choice is then made by minimizing a volume-fraction-based least-squares error $E(n)$ in L_2 and L_∞ for the 3x3 block of cells, between the predicted volume-fraction field $\tilde{C}(\mathbf{n}_n)$ in the surrounding 8 cells for the candidate n, and the actual volume fraction C. The ELVIRA method is second-order accurate, but is computationally expensive, especially in 3D where the number of normal candidates becomes large [127, 7].

For unstructured grids, Ivey and Moin [9] (2015) developed an embedded height-function framework to compute second-order normals. However, the prefered method to compute the interface normal on unstructured grids is not clear [102].

Once the normal vector is computed in Eq. (1.99), one needs to compute the constant α by enforcing liquid volume conservation. For cartesian meshes analytical methods are available [128]. For general polyhedra root-finding algorithms are typically employed: one has to find the zero of a non-linear function $f(\alpha) = V_l(\alpha) - C\Delta x^3$, with $V_l(\alpha)$ the estimated liquid volume from a guess of α, and $C\Delta x^3$ the actual liquid volume in a cell of size Δx and volume $V = \Delta x^3$ [49, 129].

1.3.1.2.2 Interface advection

The volume fraction field C is advanced in time using Eq. (1.82). The advection can be performed either direction by direction, by decomposing the time-marching procedure into a set of one-dimensional equations (split schemes), or in one step (unsplit or multi-dimensional schemes). Thus, for split advection schemes, as each of the velocity derivative components is used seperately and is generally non-zero, the third term of Eq. (1.81) is kept even if the multi-dimensional flow is incompressible [130]. The VOF transport equation thus writes:

$$\boxed{C^{n+1} = C^n - \frac{\Delta t}{V}\int_A \mathbf{F}\cdot\mathbf{dA} + \frac{\Delta t}{V}\int_V H\boldsymbol{\nabla}\cdot\mathbf{u}\,dV} \tag{1.102}$$

For unsplit advection schemes, $\boldsymbol{\nabla}\cdot\mathbf{u} = 0$ is used and hence the last term is removed.

Split advection For operator split schemes, one considers the one-dimensional version of Eq. (1.102). For sake of simplicity, the integration is performed over a 2D square cell (i, j) of area $A = \Delta x\Delta y = \Delta x^2$:

$$C_{i,j}^{n+1} = C_{i,j}^n - \frac{\Delta t}{\Delta x^2}\sum_{L_i=1}^{2} F_{L_i} + \Delta t\tilde{C}_{i,j}\frac{\partial u}{\partial x}\Big|_i \tag{1.103}$$

where $L_i = 1, 2$ are the 2D cell line boundaries with respect to direction i, crossed by the respective fluxes F_{L_i}. $\tilde{C}_{i,j}$ can be chosen either as $C^n_{i,j}$ (explicit scheme) or $C^{n+1}_{i,j}$ (implicit scheme). Using a centered finite-difference discretization for the derivative of the velocity component u along x, and considering the cell boundaries $(i+1/2, j)$ and $(i-1/2, j)$, the discretized transport equation for direction i yields:

$$C^{n+1}_{i,j} = C^n_{i,j} - \frac{\Delta t}{\Delta x^2}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) + \Delta t \tilde{C}_{i,j}\frac{u_{i+1/2,j} - u_{i-1/2,j}}{\Delta x} \tag{1.104}$$

For the Eulerian-Implicit (EI) advection scheme, $\tilde{C}_{i,j} = C^{n+1}_{i,j}$, the advection equation becomes:

$$C^{n+1}_{i,j} = \frac{C^n_{i,j} - \frac{\Delta t}{\Delta x^2}\left(F_{i+1/2,j} - F_{i-1/2,j}\right)}{1 - \frac{\Delta t}{\Delta x}\left(u_{i+1/2,j} - u_{i-1/2,j}\right)} \tag{1.105}$$

where the fluxes F are computed in an "Eulerian" manner from $C^n_{i,j}$, with an area-fluxing scheme. An onto-cell implicit linear mapping, depicted in Fig. 1.15 (a) is used, of which the results depend on the local velocity field: in the figure, a contraction is represented. The fluxes can be computed before the advection of the interface by the mapping, from the left and right shaded areas inside the tessellated rectangle of area A_x issued from the grid mapping. If we introduce a coefficient a which is the contraction/expansion coefficient of the onto-cell mapping:

$$a = \frac{1}{1 - \frac{\Delta t}{\Delta x}\left(u_{i+1/2,j} - u_{i-1/2,j}\right)} \tag{1.106}$$

The EI advection equation eventually writes:

$$\boxed{C^{n+1}_{i,j} = a\left(C^n_{i,j} - \frac{\Delta t}{\Delta x^2}\left(F_{i+1/2,j} - F_{i-1/2,j}\right)\right)} \tag{1.107}$$

Thus, the updated volume-fraction field is the sum of the compressed (or expanded) areas. If one looks at Fig. 1.15 (a) as an example: the updated C-field can be obtained either by multiplying the sum of the 3 original shaded areas in A_x on the left by the compression coefficient a, or by directly summing the compressed areas obtained after the mapping of the interface onto the central cell on the right of Fig. 1.15 (a).

For the Lagrangian-Explicit (LE) advection scheme, $\tilde{C}_{i,j} = C^n_{i,j}$, the advection equation becomes:

$$C^{n+1}_{i,j} = C^n_{i,j}\left(1 + \frac{\Delta t}{\Delta x}\left(u_{i+1/2,j} - u_{i-1/2,j}\right)\right) - \frac{\Delta t}{\Delta x^2}\left(F_{i+1/2,j} - F_{i-1/2,j}\right) \tag{1.108}$$

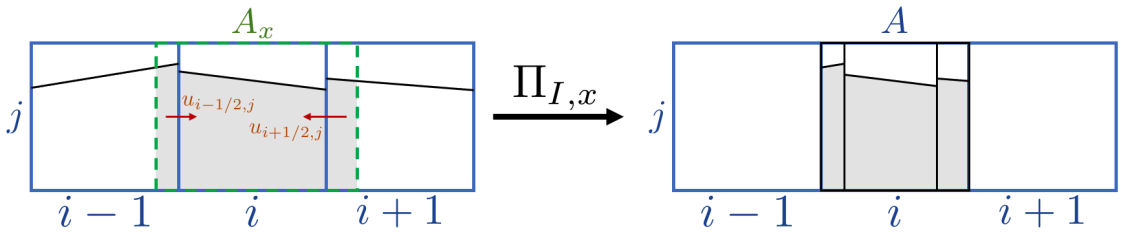
where the fluxes F are computed in an "Lagrangian" manner by advecting the end points of the interface segments. For the LE scheme an out-of-cell explicit linear mapping is used, depicted in Fig. 1.15 (b), of which the results also depend on the local velocity field. To avoid inconsistencies ($C < 0$ or $C > 1$), the fluxes are computed after the Lagrangian advection of the interface segments by the mapping [130, 131], from the left and right white areas inside the cell (i, j) in Fig. 1.15 (b) right. If we introduce a coefficient b which is the contraction/expansion coefficient of the out-of-cell mapping:

$$b = 1 + \frac{\Delta t}{\Delta x}\left(u_{i+1/2,j} - u_{i-1/2,j}\right) \tag{1.109}$$

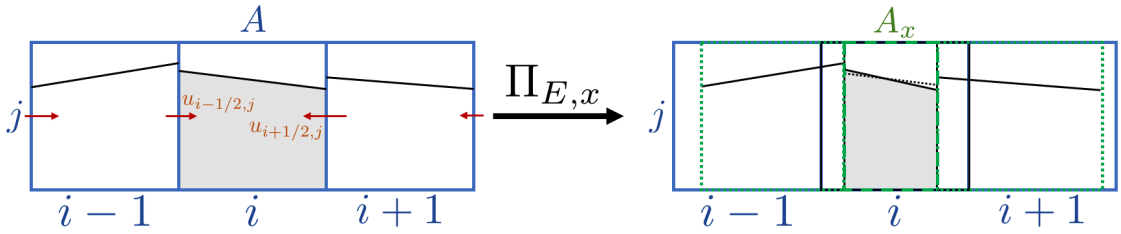
The LE advection equation eventually writes:

$$\boxed{C^{n+1}_{i,j} = bC^n_{i,j} - \frac{\Delta t}{\Delta x^2}\left(F_{i+1/2,j} - F_{i-1/2,j}\right)} \tag{1.110}$$

Thus, $bC_{i,j}^n$ is the compressed shaded area inside the compressed central cell of area A_x in Fig. 1.15 (b) right. As mentioned in [125], the Lagrangian advection by a one-dimensional flow of a VOF/-PLIC reconstruction changes its orientation, as shown by the black dotted interface line in Fig. 1.15 (b) right. Therefore, the EI and LE fluxes are different.



(a) Onto-cell implicit linear mapping



(b) Out-of-cell explicit linear mapping

Figure 1.15: Illustration of the mappings used in the split advection schemes: (a) Contraction using a horizontal onto-cell implicit linear mapping $\Pi_{I,x}$; (b) Contraction using a horizontal out-of-cell explicit linear mapping $\Pi_{E,x}$.

To achieve a great conservation property, the two schemes presented above are usually combined to give the well-known EI-LE advection scheme. Indeed, the implicit mapping sequence along x followed by the explicit one along y conserves area (or mass) exactly. This 2D combination can be either split if an intermediate reconstruction of the interface is performed between the two direction sweeps, or unsplit. The discrete incompressibility constraint $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$ writes:

$$\left(u_{i+1/2,j} - u_{i-1/2,j}\right) + \left(v_{i,j+1/2} - v_{i,j-1/2}\right) = 0 \tag{1.111}$$

The combined linear mapping Π multiplies the area A_x of the initial rectangle in Fig. 1.15 (a) by the product of the compression/expansion coefficients ab:

$$ab = \frac{1 + \frac{\Delta t}{\Delta x}\left(v_{i,j+1/2} - v_{i,j-1/2}\right)}{1 - \frac{\Delta t}{\Delta x}\left(u_{i+1/2,j} - u_{i-1/2,j}\right)} \tag{1.112}$$

which equals to 1 by using Eq. (1.111), leading to exact area conservation:

$$A_y = \Pi\left(A_x\right) = \Pi_{E,y}\left(\Pi_{I,x}(A_x)\right) = abA_x = A_x \tag{1.113}$$

To avoid asymmetries, the direction of the first implicit advection should be alternated in time [49, 125], e.g.:

$$\begin{cases} t^n \rightarrow t^{n+1} : x - y \\ t^{n+1} \rightarrow t^{n+2} : y - x \end{cases} \tag{1.114}$$

The 2D EI-LE split scheme is here summarized:

(i) Reconstruct the interface Γ at t^n from C^n;

(ii) Compute the Eulerian area fluxes in the first spatial direction;

(iii) Update an intermediate volume fraction field ($C^n \rightarrow C^*$) using the EI scheme;

(iv) Reconstruct the intermediate interface Γ^* from the intermediate volume fraction field C^*;

(v) Compute the Lagrangian fluxes in the other spatial direction;

(vi) Update the volume fraction field ($C^* \rightarrow C^{n+1}$) using the LE scheme;

(vii) Reconstruct the interface Γ^{n+1} from the updated volume fraction field C^{n+1}.

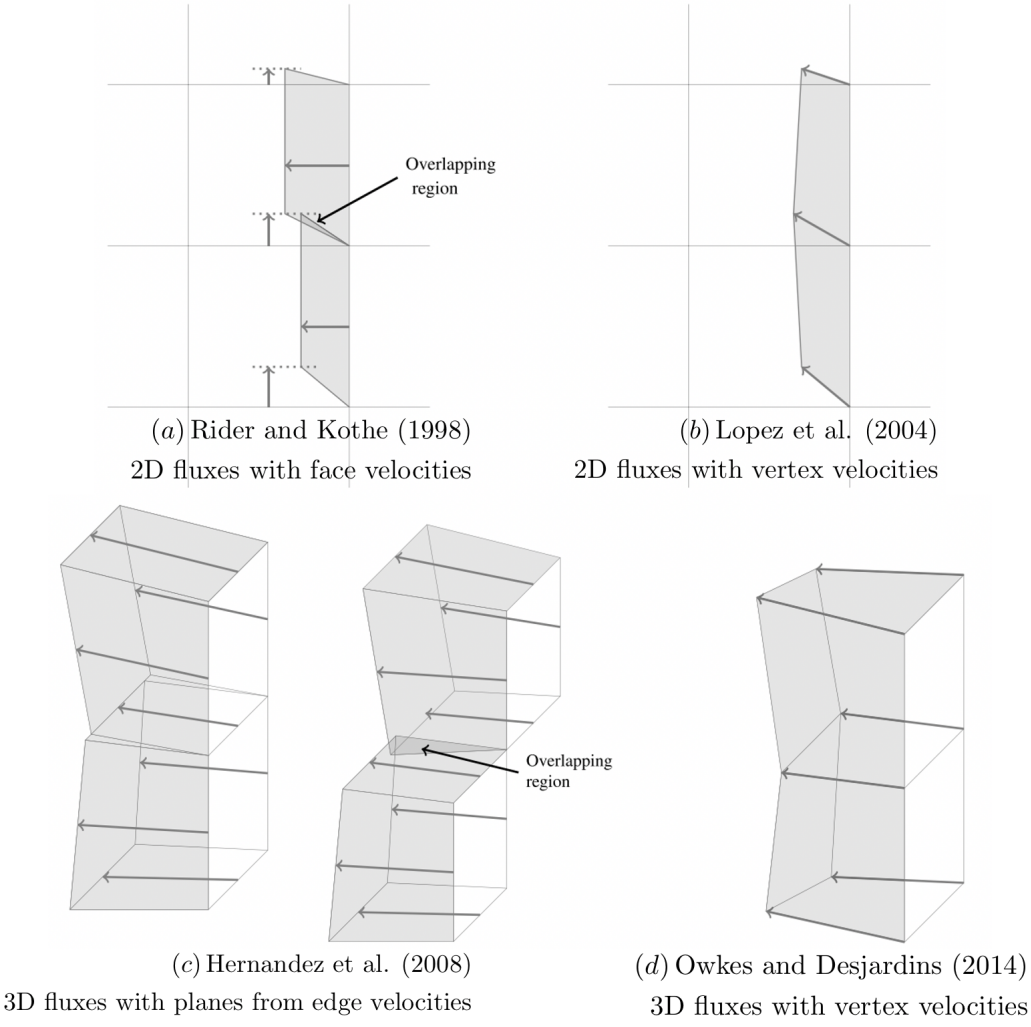
The first EI-LE scheme which is bounded (i.e. $0 \leq C \leq 1$) and mass-conserving has been proposed in 2D in [124, 132] (2003). This bounded and mass-conserving method has then been extended to 3D in [125] (2007), and consists of 3 EI-LE advection steps (one for each spatial direction x, y, z), corresponding to 6 advection/reconstruction steps in total.

In 2010, Weymouth and Yue [133] proposed a split advection scheme, conditionally bounded and discretely mass-conserving, which consists of three LE steps. The boundedness property is ensured by the mean of a CFL restriction criterion.

Unsplit advection For unsplit advection schemes, the transported area (2D) or volume (3D) is fluxed in one multi-dimensional step. These schemes are accurate but their implementation can become tricky, as complex flux polyhedra in computational cells containing the interface need to be computed. This is particularly true in 3D and with unstructured meshes. Moreover, additional treatments are necessary to address mass conservation issues in 3D.

One of the first Eulerian unsplit scheme is the one of Rider and Kothe (1998) [130], proposed for 2D cartesian meshes. In this method, the face-based velocity components on a staggered MAC grid are used to compute the trapezoidal flux, which accounts for the multidimensionality of the flow. This scheme leads to overlapping region and hence mass conservation issues to tackle. To address this issue, Lopez et al. (2004) [134] proposed a scheme, still for 2D cartesian grids, which uses cell-vertex velocities instead, computed by interpolating the face-based velocities. With this construction, no overlap regions appear and thus mass conservation properties are improved compared to the scheme of Rider and Kothe.

For 3D cartesian meshes, Hernandez et al. (2008) [135] proposed an extension of the work of Rider and Kothe, using planes from edge-based velocities to compute the volume fluxes. As in Rider and Kothe (1998), this method can lead to overlapping regions and hence mass conservation issues. Later, Owkes and Desjardins (2014) [136] introduced a method which can be seen as an extension of the works of Lopez et al. (2D with vertex velocities) and Hernandez et al. (3D with edge velocities), with discrete mass conservation. As in Lopez et al., the fluxes are based on vertex velocities. This scheme is a bounded, mass-conserving, 3D unsplit VOF advection algorithm. Fig. 1.16, reprinted from [136], illustrates the flux computation in the 2D and 3D cartesian unsplit advection schemes mentioned. Owkes and Desjardins further advanced their unsplit advection scheme in [137] (2017), by adding a refined grid that provides consistent fluxes of mass and momentum defined on a staggered grid and discrete conservation of mass and momentum, even for flows with large density ratios. This modification has then been applied on the electrohydrodynamic-assisted atomization of a liquid jet, for which the PLIC interface is shown in Fig. 1.17.



(*a*) Rider and Kothe (1998)
2D fluxes with face velocities

(*b*) Lopez et al. (2004)
2D fluxes with vertex velocities

(*c*) Hernandez et al. (2008)
3D fluxes with planes from edge velocities

(*d*) Owkes and Desjardins (2014)
3D fluxes with vertex velocities

Figure 1.16: Illustration of the flux computation in various cartesian unsplit advection schemes. This figure is reprinted from [136].

Regarding unstructured grids, Jofre et al. (2014) [7] extended the scheme of Owkes and Desjardins (2014), with the treatment of non-convex polyhedra. Later, Ivey and Moin (2017) [10] improved the computation of flux polyhedra to propose a discretely conservative and bounded unsplit VOF advection scheme on unstructured meshes.

1.3.2 Level Set (LS)

1.3.2.1 Standard Level Set (SLS)

Classical level set methods, introduced by Sethian (1982) [91, 92] is a powerful tool to track evolving interfaces in computational geometry, and rely on the use of a smooth, continuous function $\phi(\mathbf{x}, t)$. This function is a signed-distance function to the interface Γ:

$$\phi(\mathbf{x}, t) = \pm|\mathbf{x}(t) - \mathbf{x}_\Gamma(t)| \tag{1.115}$$

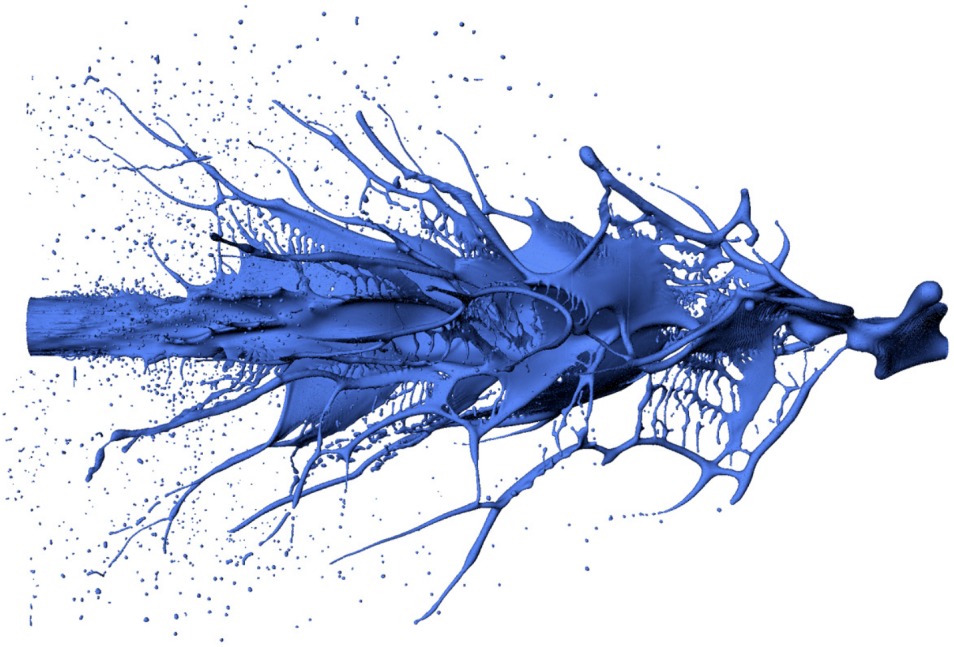


Figure 1.17: Electrohydrodynamic-assisted atomization of a liquid jet. Reprinted from [137].

where $\mathbf{x}_\Gamma(t)$ is the position of the interface at time t. For liquid-gas flows, the sign of ϕ is determined by the fluid phase, e.g.:

$$\begin{cases} \phi(\mathbf{x},t) > 0 \text{ if } \mathbf{x} \text{ is in liquid} \\ \phi(\mathbf{x},t) = 0 \text{ if } \mathbf{x} \text{ is at the interface } (\mathbf{x} = \mathbf{x}_\Gamma) \\ \phi(\mathbf{x},t) < 0 \text{ if } \mathbf{x} \text{ is in gas} \end{cases} \tag{1.116}$$

Thus, the interface is the zero-level implicit surface of ϕ:

$$\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^3 | \, \phi(\mathbf{x},t) = 0\} \tag{1.117}$$

A one-dimensional representation which illustrates the principle of the standard level set method is given by Fig. 1.18.

Even though one only describes the application of traditional level set methods to non-reacting liquid-gas flow problems in this thesis, they can be used for a wide range of applications, such as computer vision, combustion, shock propagation, electromigration, solidification, optimal path planning, or grid generation. Level set techniques have been extended to two-phase flows by Sussman et al. (1994) [4].

The level set function moves with the fluid, at the flow velocity $\mathbf{u}$:

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla} \phi = 0 \tag{1.118}$$

If the unit interface normal $\mathbf{n} = \boldsymbol{\nabla}\phi / |\boldsymbol{\nabla}\phi|$ is introduced, the transport equation becomes:

$$\boxed{\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \mathbf{n} |\boldsymbol{\nabla}\phi| = 0} \tag{1.119}$$

However, ϕ loses its signed-distance property after advection: where the interface is stretched, the gradient of ϕ becomes steeper, and where the interface is compressed the gradient becomes

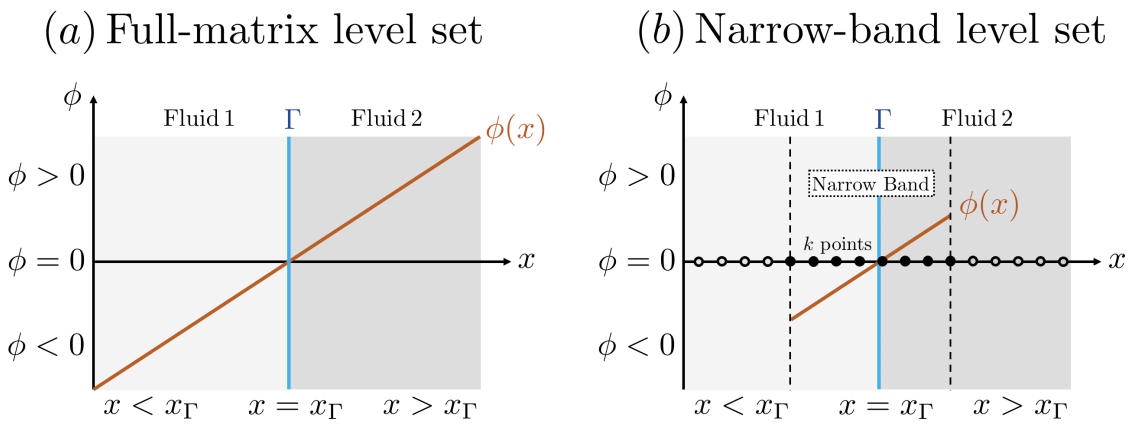


Figure 1.18: One-dimensional illustration of the standard level set method: (a) The level set is computed in the entire domain; (b) The level set is computed only in the neighborhood of the interface Γ, here with $k = 4$ points on each side of Γ (adapted from [92]). In this work: "Fluid 1" is gas, "Fluid 2" is liquid.

smoother. To fix this problem, Sussman et al. (1994) [4] introduced a reinitialization procedure to make ϕ a signed-distance function again:

$$\boxed{\frac{\partial \phi}{\partial \tau} + \mathrm{sgn}(\phi_0)(|\boldsymbol{\nabla}\phi| - 1) = 0} \tag{1.120}$$

with ϕ_0 the level set function after transport that needs to be reinitialize, and $\mathrm{sgn}(\phi_0)$ the signature function of ϕ_0:

$$\mathrm{sgn}(\phi_0) = \begin{cases} 1 & \text{if } \phi_0 > 0 \\ 0 & \text{if } \phi_0 = 0 \\ -1 & \text{if } \phi_0 < 0 \end{cases} \tag{1.121}$$

Eq. (1.120) is a Hamilton-Jacobi equation, with the Hamiltonian $\mathcal{H} = \mathrm{sgn}(\phi_0)(|\boldsymbol{\nabla}\phi| - 1)$, solved in pseudo-time τ until convergence to $|\boldsymbol{\nabla}\phi| = 1$, which yields a signed-distance function. This ensures that the slope of the function near the interface remains unchanged over time. The reinitialization step is critical in all level-set based methods, as it can introduce spurious deformation of the interface, potentially leading to very unphysical solutions and bad mass-conservation properties. Thus, improvements have been brought to the original reinitialization step. For example, the sign function $\mathrm{sgn}(\phi_0)$ can be replaced by a smoother function $S(\phi)$, as suggested by Peng et al. (1999) [138]:

$$\frac{\partial \phi}{\partial \tau} + S(\phi)(|\boldsymbol{\nabla}\phi| - 1) = 0 \tag{1.122}$$

where:

$$S(\phi) = \frac{\phi}{\sqrt{\phi^2 + |\boldsymbol{\nabla}\phi|^2 \Delta x^2}} \tag{1.123}$$

An advantage of traditional level set methods is the accuracy they provide to compute interface topology properties, such as normals $\mathbf{n} = \boldsymbol{\nabla}\phi/|\boldsymbol{\nabla}\phi|$ and curvature $\kappa = -\boldsymbol{\nabla} \cdot \mathbf{n}$ due to the smoothness of the ϕ-field. Classical level set methods are rather simple, efficient, but not mass conserving, and hence considerable efforts have been put into improving the original method, leading to various techniques which derive from the original framework presented above. These

improvements come however at the cost of additional complexity, and hence taking care of the computational performance of the methods becomes critical.

To this aim, Chopp (1993) [139], then Adalsteinsson and Sethian (1995) [36] proposed to work only near the zero-contour of ϕ, i.e. the front of interest. This process is called "narrow banding". Using a narrow band around the interface reduces the number of operations per iteration, and is thus a rather straightforward way to implement fast level set methods. Indeed, in 3D, the number of operations in one time step lowers from $\mathcal{O}(N^3)$ to $\mathcal{O}(2kN^2)$, with N the number of grid points in one direction and $2k$ the total number of points in the narrow band, if one considers that the front has roughly N^2 points [92]. This approach can also be beneficial regarding the explicit CFL time-step restriction: the criterion can be defined only in the narrow band instead of looking at the maximum advection velocity in the whole computational domain.

Fast numerical methods, such as the Fast Marching Method (FMM) of Chopp (2001) [140] or the Fast Sweeping Method (FSM) of Zhao (2005) [141], can also be used for the reinitialization step to solve the Eikonal equation $|\boldsymbol{\nabla}\phi| = 1$. These methods are efficient and accurate, but their parallel implementation is challenging [102].

1.3.2.2 Conservative Level Set (CLS)

To reduce the mass losses of the original level set method, several authors have used the fact that these losses reduce under mesh refinement (spatial convergence of the mass error). For example Herrmann (2008) [15] uses an auxiliary fine cartesian grid to resolve the level set field, while Gibou et al. (2018) [142] use octree Adaptive Mesh Refinement (AMR) close to the interface. These high-resolution level-set approaches indeed reduce mass conservation errors, which depend on the two-phase flow simulated, but do not allow mass conservation to within machine accuracy. For instance, when simulating a complex turbulent atomizing spray, droplets at resolution limit or poorly-resolved will inevitably be present in the domain, and can thus still disappear.

An attempt towards discrete mass conservation is the Conservative Level Set (CLS) method of Olsson and Kreiss (2005) [29], which has emerged from a smeared out Heaviside function $\tilde{H}$, originally computed from the signed-distance function ϕ to reconstruct the material properties (density, viscosity):

$$\tilde{H}(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2} + \frac{\phi}{2\varepsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right) & \text{if } -\varepsilon \leq \phi \leq \varepsilon \\ 1 & \text{if } \phi > \varepsilon \end{cases} \tag{1.124}$$

where ε is half the thickness of the interface profile. Similarly, Olsson and Kreiss (2005) used a continuous smoothed Heaviside function ψ to represent the interface:

$$\psi(\mathbf{x}, t) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\mathbf{x}, t)}{2\varepsilon}\right) + 1\right) \tag{1.125}$$

with the interface normals and curvature computed as:

$$\begin{cases} \mathbf{n} = \frac{\boldsymbol{\nabla}\psi}{|\boldsymbol{\nabla}\psi|} \\ \kappa = -\boldsymbol{\nabla}\cdot\mathbf{n} = -\boldsymbol{\nabla}\cdot\left(\frac{\boldsymbol{\nabla}\psi}{|\boldsymbol{\nabla}\psi|}\right) \end{cases} \tag{1.126}$$

Using ψ, the interface is located at the iso-level $1/2$:

$$\Gamma(t) = \{\mathbf{x} \in \mathbb{R}^3 |\, \psi(\mathbf{x}, t) = 1/2\} \tag{1.127}$$

Assuming the flow is incompressible, the scalar ψ is advected as:

$$\boxed{\frac{\partial \psi}{\partial t} + \boldsymbol{\nabla} \cdot (\psi \mathbf{u}) = 0} \tag{1.128}$$

and then reshaped to its original hyperbolic-tangent shape using the following reinitialization equation, integrated in pseudo-time:

$$\frac{\partial \psi}{\partial \tau} = \boldsymbol{\nabla} \cdot \left(\underbrace{\varepsilon \boldsymbol{\nabla} \psi}_{\text{Isotropic diffusion flux}} - \underbrace{\psi(1-\psi)\mathbf{n}}_{\text{Normal resharpening flux}} \right) \tag{1.129}$$

Later, Olsson et al. (2007) [30] modified the reinitialization Eq. (1.129) by projecting the diffusion flux in the normal direction:

$$\frac{\partial \psi}{\partial \tau} = \boldsymbol{\nabla} \cdot \left(\underbrace{\varepsilon (\boldsymbol{\nabla} \psi \cdot \mathbf{n}) \mathbf{n}}_{\text{Normal diffusion flux}} - \underbrace{\psi(1-\psi)\mathbf{n}}_{\text{Normal resharpening flux}} \right) \tag{1.130}$$

which reduces the errors induced by tangential fluxes.

The CLS method is not strictly conservative for two reasons. First, the approximate and exact enclosed volumes of liquid are respectively:

$$\widetilde{V}_l = \int_V \psi \, \mathrm{d}V \quad \text{and} \quad V_l = \int_V H(\psi - 1/2) \, \mathrm{d}V \tag{1.131}$$

where H is the Heaviside function (0 in gas, 1 in liquid). In the CLS method, the approximate volume $\widetilde{V}_l$ is conserved, but differs from the exact volume V_l due to interface curvature, i.e. in the ubiquitous case of non-straight interfaces. Moreover, other sources of numerical errors arise on non-uniform meshes, when the cell size varies locally: this point will be discussed in the next chapter.

Desjardins et al. (2008) [31] further improved the CLS method, yielding the Accurate Conservative Level Set (ACLS) method. In order to avoid spurious normal orientation due to the sharp nature of ψ, they proposed to compute the normal vector $\mathbf{n}$ using a reconstructed signed-distance function ϕ with the help of a Fast Marching Method (FMM):

$$\mathbf{n} = \frac{\boldsymbol{\nabla} \phi_{\mathrm{FMM}}}{|\boldsymbol{\nabla} \phi_{\mathrm{FMM}}|} \tag{1.132}$$

This method has been used to simulate the turbulent atomization of a liquid Diesel jet, as depicted in Fig. 1.19.

However, solving Eq. (1.130) can lead to dispersion errors. Hence, in order to keep ψ bounded, TVD schemes have to be employed, resulting in numerical dissipation. Moreover, high levels of reinitialization using Eq. (1.130) result in artificial deformation of the interface, even if the normal vector is exact, as detailed in McCaslin and Desjardins (2014) [143].

To limit spurious deformation of the interface, especially when the interface is (quasi-)static, McCaslin and Desjardins (2014) [143] proposed a localized reinitialization process, to reinitialize according to need, by introducing a spatially and temporally varying coefficient α:

$$\frac{\partial \psi}{\partial \tau} = \boldsymbol{\nabla} \cdot \left(\alpha \left(\underbrace{\varepsilon (\boldsymbol{\nabla} \psi \cdot \mathbf{n}) \mathbf{n}}_{\text{Normal diffusion flux}} - \underbrace{\psi(1-\psi)\mathbf{n}}_{\text{Normal resharpening flux}} \right) \right) \tag{1.133}$$

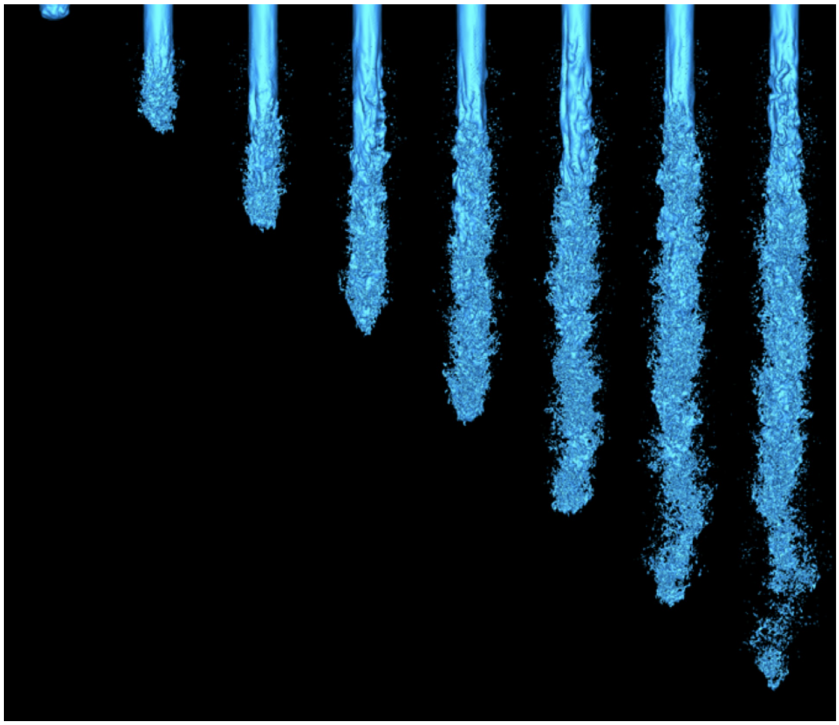


Figure 1.19: Turbulent atomization of a liquid Diesel jet with the original Accurate Conservative Level Set (ACLS) method (2008). Reprinted from [31].

which is included in the pseudo-time integration $\tilde{\tau} = \alpha\tau$ after mathematical developments which can be found in [143]:

$$\frac{\partial \psi}{\partial \tilde{\tau}} = \boldsymbol{\nabla} \cdot \left(\underbrace{\varepsilon(\boldsymbol{\nabla}\psi \cdot \mathbf{n})\mathbf{n}}_{\text{Normal diffusion flux}} - \underbrace{\psi(1-\psi)\mathbf{n}}_{\text{Normal resharpening flux}} \right) \tag{1.134}$$

As mentioned in their work, the coefficient α should be defined in a way that reflects how much reinitialization is locally required. The amount of necessary reinitialization must depend on how much the level set has been locally deformed. Two sources of level set deformation are identified in their work: (i) the kinematics of the velocity field and (ii) numerical diffusion associated with the evolution of ψ through the discrete solution of the transport equation (Eq. (1.128)). Their method leads to a reduction of spurious interface deformation for the less active parts of the interface, as displayed in Fig. 1.20.

Waclawczyk (2015) [144] treated the issue of artificial deformation of the interface by taking advantage of an analytical signed-distance level set to find a reinitialization equation similar to the traditional Hamilton-Jacobi level-set reinitialization equation Eq. (1.120) as:

$$\frac{\partial \psi}{\partial \tau} = \boldsymbol{\nabla} \cdot \Big(\psi(1-\psi) \left(|\boldsymbol{\nabla}\phi_{\text{map}} \cdot \mathbf{n}| - 1 \right) \mathbf{n} \Big) \tag{1.135}$$

where $\phi_{\text{map}} = \varepsilon \ln\left(\psi/(1-\psi)\right)$ is an analytical signed-distance function, mapped in the transient part of the hyperbolic tangent profile: i.e. $\psi \in]0;1[$. In [144], the normal vector is computed at the beginning of the reinitialization iteration process from this mapped distance function: $\mathbf{n} = \boldsymbol{\nabla}\phi_{\text{map},\tau=0}/|\boldsymbol{\nabla}\phi_{\text{map},\tau=0}|$. This idea then paved the way to Chiodi and Desjardins (2017) [35], who proposed a reformulation of the sharp term $\psi(1-\psi)$, leading to a form of the reinitialization

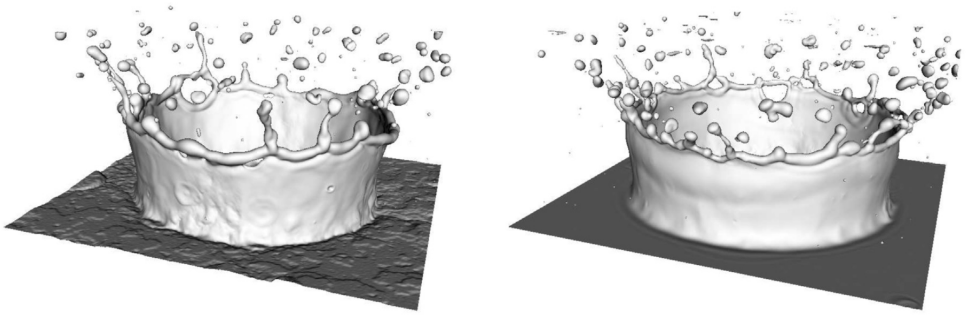


Figure 1.20: Outcome of a drop impact onto a static free surface; Left: uniform reinitialization, Right: Localized reinitialization. Reprinted from [143].

which is very accurate, and thus does not induce large deformation of the interface, even at high levels of reinitialization. This is very significant when looking at the case of static interfaces. This reformulated reinitialization writes:

$$\boxed{\frac{\partial \psi}{\partial \tau} = \boldsymbol{\nabla} \cdot \left(\frac{1}{4 \cosh^2 \left(\frac{\phi_{\text{map}}}{2\varepsilon} \right)} \left(|\boldsymbol{\nabla} \phi_{\text{map}} \cdot \mathbf{n}| - 1 \right) \mathbf{n} \right)} \tag{1.136}$$

with the normals computed using Eq. (1.132). The flux is now written without ψ, and is thus less sensitive to its bounds. As a consequence, non-TVD schemes can be employed. In [35], this upgraded ACLS method has been used to simulate a turbulent liquid jet being injected into quiescent gas, demonstrating the capabilities of the method to simulate complex turbulent three-dimensional liquid-gas flows. The results are reprinted in Fig. 1.21. However, similarly to purely compressive

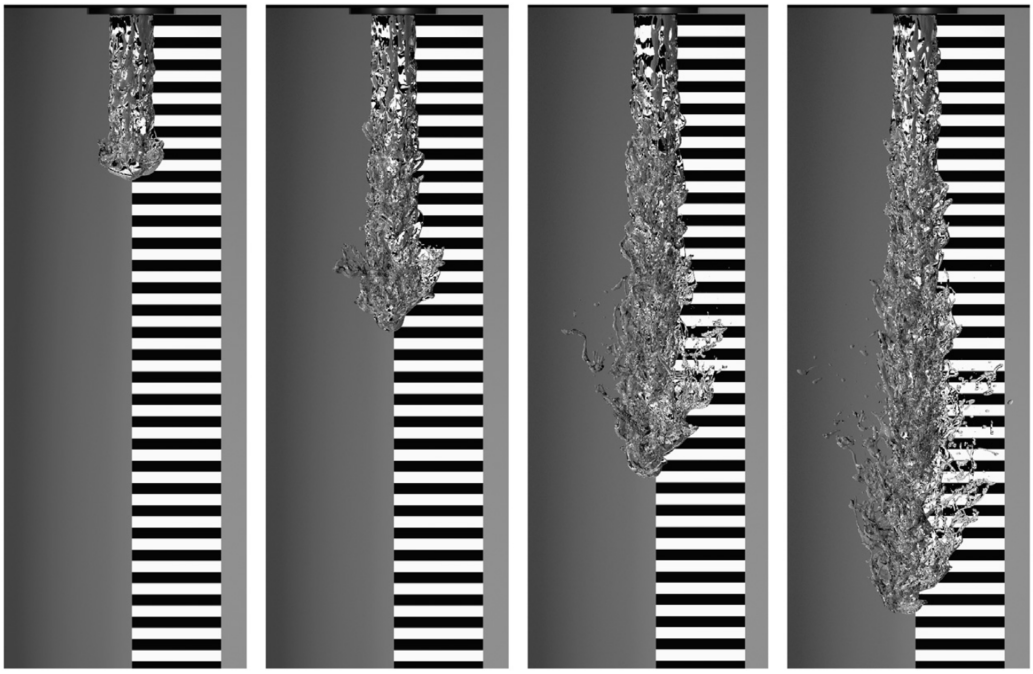


Figure 1.21: DNS of a turbulent liquid jet injected into quiescent gas using the reformulated ACLS method (2017): renderings of the liquid-gas interface. Reprinted from [35].

schemes in Section 1.3.1.1.1, this form is very "compressive": it is susceptible to induce locking

on the mesh. In this case the interface can be aligned with grid edges, potentially leading in turn to bad prediction of implicit topology changes. This equation can be further improved to solve this "locking issue". This particular point is treated in this thesis: a locking fix is proposed, improving for example front-merging scenarios. An extension of the ACLS method using the reinitialization Eq. (1.130) to unstructured three-dimensional grids can be found in Balcazar et al. (2014) [17]. In the present research, an extension of the ACLS method to 3D unstructured grid is also proposed, featuring the state-of-the-art reinitialization process: Eq. (1.136), which to our knowledge has never been implemented for unstructured grids prior to this work. Nonetheless, the locking fix mentioned above can be used for both cartesian and unstructured meshes.

1.3.3 Coupled Level-Set and Volume-Of-Fluid method (CLSVOF)

The CLSVOF method is a hybrid interface-capturing procedure, combining the VOF method for its mass-conservation properties together with the classical Level Set for an accurate computation of interface-topology features (normals and curvature). The overall accuracy of the method is higher than VOF or LS methods separatly, but is more computationally expensive. The first CLSVOF method has been introduced by Sussman and Puckett (2000) [145]. One can also mention the CLSVOF method of Ménard et al. [146], which led to one of the first detailed simulations of an atomizing liquid jet: see Fig. 1.22. This latter method is implemented in the well-known Archer DNS code, also developed at CORIA laboratory by Berlemont and collaborators. Literature results, obtained prior to the present work with the Archer code, are used in this thesis to validate some of the results presented in Chapter 4.

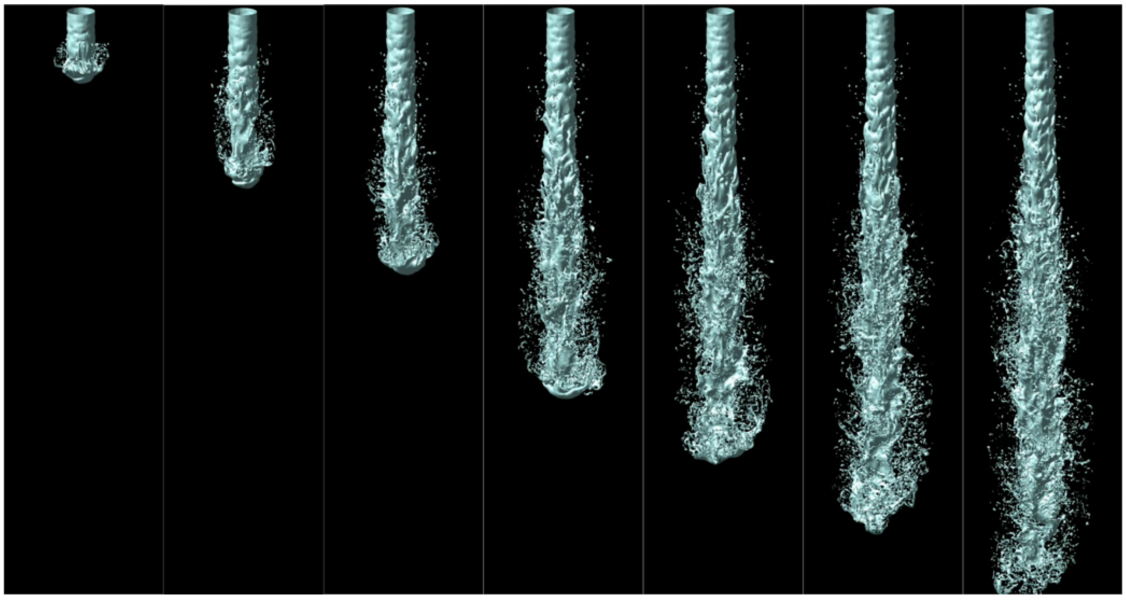


Figure 1.22: Development of an atomizing liquid jet using the CLSVOF method, implemented in the Archer DNS code. Reprinted from [146].

1.3.4 Moment Of Fluid methods (MOF)

The Moment Of Fluid method (MOF), originally introduced in [147, 148, 149] and then applied to multiphase flow simulations in [150, 151, 152], is an extension of the geometric VOF method,

resulting in improved accuracy. In addition to the liquid volume fraction C (0th moment of liquid volume), the reference phase centroid (or center of mass) within a computational cell $\mathbf{x}_l$ (1st moment of liquid volume) is also tracked:

$$\mathbf{x}_l = \frac{1}{V}\int_V \mathbf{x} H \, dV \tag{1.137}$$

where V is the control volume, and H the Heaviside function (1 in liquid, 0 in gas). These data are sufficient to reconstruct the approximate interface within each mixed cell (i.e. $0 < C < 1$), and thus no additional data are required from the neighboring cells [152]. The MOF method reconstructs any linear interface exactly and is thus second-order accurate [147, 148, 152].

The MOF method with PLIC interface reconstruction ($\mathbf{n}\cdot\mathbf{x}+\alpha = 0$, see Eq. (1.99)) necessitates the determination of interface normal $\mathbf{n}$ and shortest distance α of interface from cell center, as in geometric VOF methods. While α is computed by enforcing liquid volume conservation as in VOF methods (i.e. volume fraction error $E_0 = 0$), the normal $\mathbf{n}$ is determined simultaneously from a constrained minimization of centroid defect E_1 [150, 152]:

$$\begin{cases} E_0(\mathbf{n},\alpha) = |\tilde{C}(\mathbf{n},\alpha) - C| = 0 \\ E_1(\mathbf{n},\alpha) = \left|\left|\tilde{\mathbf{x}}_l(\mathbf{n},\alpha) - \mathbf{x}_l\right|\right|_2 \end{cases} \tag{1.138}$$

where, similarly to the ELVIRA technique [126] for geometric VOF algorithms (see Section 1.3.1.2.1), the predicted field $\tilde{.}$ given the guessed plane reconstruction $(\mathbf{n},\alpha)$ is compared to the original (reference) field. An illustration of the MOF-PLIC interface reconstruction is displayed in Fig. 1.23.

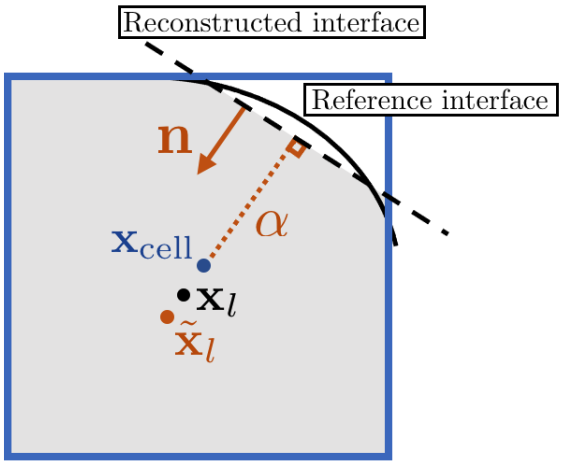


Figure 1.23: Illustration of the MOF-PLIC interface reconstruction using liquid centroids. Inspired by [152].

In MOF methods, the volume fraction and the phase centroids are advected simultaneously and consistently in each cartesian direction using split algorithms: for instance in [152] the EI-LE scheme [124, 132, 125] for phase centroids together with the Weymouth-Yue procedure [133] for the volume fraction have been used (see Section 1.3.1.2.2).

It should be noted that the MOF method is computationally expensive compared to the other existing interface-capturing methods, which motivates the use of hybrid techniques to find a compromise between accuracy and cost [153]. In Mukundan et al. (2022) [153], a switch is performed,

based on local interface curvature resolution, between MOF and CLSVOF methods throughout the simulations: the under-resolved regions are treated using the MOF method. This hybrid MOF-CLSVOF technique, applied to the simulation of an atomizing liquid diesel jet, is depicted in Fig. 1.24.

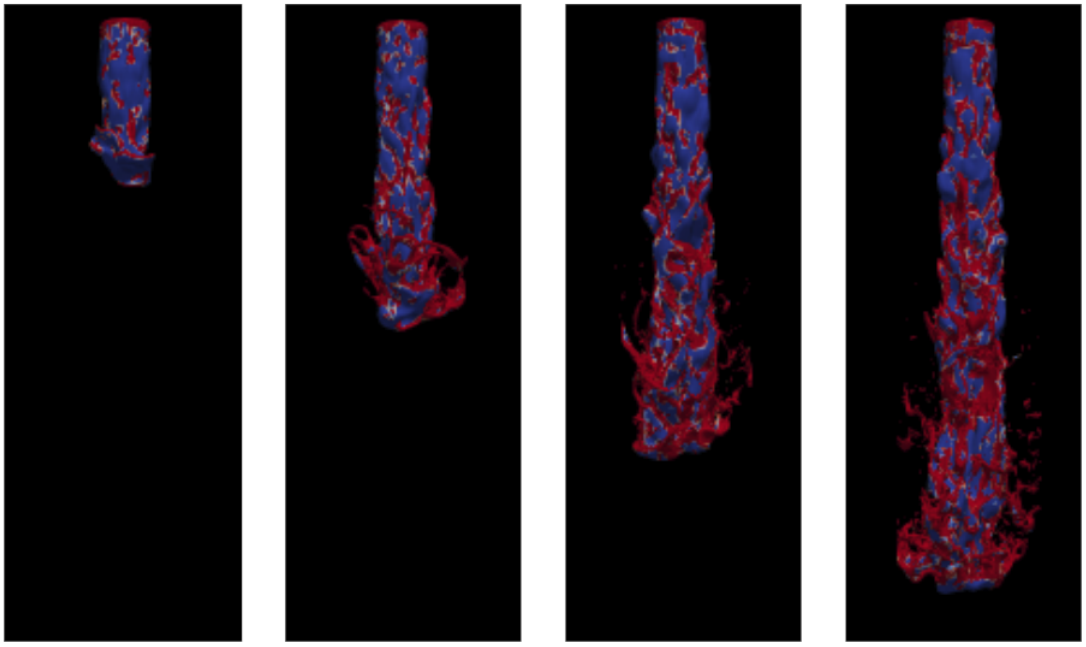


Figure 1.24: MOF-CLSVOF hybridation technique applied to the atomization of a turbulent liquid diesel jet. Red regions are treated using the MOF interface reconstruction, while blue regions employ the CLSVOF method. Reprinted from [153].

1.4 Surface-tension modeling methods

After having discussed the main sharp interface-capturing methods as a numerical tool to compute the motion of a material interface, one now needs, for liquid-gas flows, to compute surface tension forces in the sharp-interface framework. According to Popinet (2018) [50], the surface-tension models can be classified into two categories: (i) integral and (ii) volumetric numerical formulations. For (i), an expression of the integrated surface tension force is written, which is found to be advantageous in terms of momentum conservation in finite-volume frameworks. Moreover, these formulations do not require the computation of the interface curvature, which is one of the bottlenecks of numerical methods for liquid-gas flows. For (ii), a local force per unit volume (i.e. volumetric) is considered, which depends on the curvature of the interface κ, and on a numerical approximation to regularize the sharp Heaviside function H. Thus, all volumetric methods lead to a characteristic interface thickness. The volumetric methods have proven to be robust over the years, and are the most widely used in the literature to simulate 3D complex atomizing jets, which is the goal of the present work.

1.4.1 Surface tension formulations

1.4.1.1 Continuum Surface Stress (CSS) - Integral formulations

The Continuum Surface Stress (CSS) method has been introduced by Gueyffier et al. [154] (1999). One can write the volumetric surface tension force $\mathbf{F}_\sigma$ with a constant surface tension coefficient, introduced earlier in this chapter (see Eq. (1.16)), as:

$$\mathbf{F}_\sigma = \sigma\kappa\mathbf{n}\delta_A = \boldsymbol{\nabla}\cdot\mathbf{T}_{\sigma,A}\delta_A \tag{1.139}$$

with $\mathbf{T}_{\sigma,A}$ the surface stress tensor associated with surface tension. This tensor is defined as [49]:

$$\mathbf{T}_{\sigma,A} = \sigma\mathbf{I}_A = \sigma(\mathbf{I}-\mathbf{n}\otimes\mathbf{n}) \tag{1.140}$$

The volumetric surface tension force is then expressed as:

$$\mathbf{F}_\sigma = \boldsymbol{\nabla}\cdot\Big(\sigma(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})\delta_A\Big) \tag{1.141}$$

For a VOF method, the surface Dirac delta function δ_A can be approximated as $|\boldsymbol{\nabla}C|$ (this approximation will be detailed in the subsequent section), which eventually yields:

$$\boxed{\mathbf{F}_\sigma = \boldsymbol{\nabla}\cdot\Big(\sigma(\mathbf{I}-\mathbf{n}\otimes\mathbf{n})|\boldsymbol{\nabla}C|\Big)} \tag{1.142}$$

The expression above has two main advantages: it does not require to compute a curvature, and it is inherently conservative. Indeed, this method is related to the integral formulation of surface tension. By noticing that $\mathbf{n} = \mathbf{t}_1\otimes\mathbf{t}_2$, with $\mathbf{t}_{1,2}$ two orthonormal tengent vectors to the surface, the volumetric force writes [49]:

$$\mathbf{F}_\sigma = \boldsymbol{\nabla}\cdot\Big(\sigma(\mathbf{t}_1\otimes\mathbf{t}_1+\mathbf{t}_2\otimes\mathbf{t}_2)|\boldsymbol{\nabla}C|\Big) \tag{1.143}$$

If this equation is integrated in the context of a finite-volume method:

$$\int_V \mathbf{F}_\sigma\, dV = \int_V \boldsymbol{\nabla}\cdot\Big(\sigma(\mathbf{t}_1\otimes\mathbf{t}_1+\mathbf{t}_2\otimes\mathbf{t}_2)|\boldsymbol{\nabla}C|\Big)\, dV = \oint_A \Big(\sigma(\mathbf{t}_1\otimes\mathbf{t}_1+\mathbf{t}_2\otimes\mathbf{t}_2)|\boldsymbol{\nabla}C|\Big)\cdot\mathbf{dA} \tag{1.144}$$

Thus, as mentioned in [50], the contributions of surface tension forces to neighboring control volumes cancel out exactly which ensures exact local and global momentum conservation for surface tension. The CSS method however, requires the numerical approximation of the surface Dirac function δ_A, leading to a characteristic interface thickness.

Another method which is related to the integral formulation of surface tension and that does not introduce any characteristic interface thickness is the momentum-conservative and well-balanced surface tension model of Abu-Al-Saud et al. [52] (2018). This model, introduced in 2D using a level set representation of the interface, uses Eqs. (1.20) and (1.21) to write the integral surface tension force between two points A and B at which the interface intersects a control volume:

$$\boxed{\int_V \mathbf{F}_\sigma\, dV = \oint_A^B \sigma\,\mathbf{dt} = \sigma(\mathbf{t}_B-\mathbf{t}_A)} \tag{1.145}$$

The surface tension force is hence computed by the difference between entry and exit tensions, leading to an intrinsically-conservative model.

Note that the approaches presented in this section naturally include Marangoni effects, which is not the case for volumetric formulations, where the surface gradient term needs to be added in $\mathbf{F}_\sigma$ in case of variable surface tension coefficient.

1.4.1.2 Continuum Surface Force (CSF)

The Continuum Surface Force (CSF) method for modeling surface tension is one of the most popular (if not the most) model, especially for VOF interface-capturing schemes. It belongs to volumetric formulations of surface tension, which can all be summarized as [50]:

$$\mathbf{F}_\sigma = \sigma\kappa\mathbf{n}\delta_A = \sigma\kappa\boldsymbol{\nabla}H(\mathbf{x}-\mathbf{x}_\Gamma) \tag{1.146}$$

The idea of this class of methods is to find a proper numerical approximation H_ε of the Heaviside function H, which leads to a characteristic interface thickness ε so that:

$$\lim_{\varepsilon\to 0} H_\varepsilon = H \tag{1.147}$$

In the CSF method, introduced by Brackbill et al. [155] (1992) for VOF methods, the volume fraction field C is used: $H_\varepsilon = C$, with $\varepsilon = \Delta x$ the cell size:

$$\boxed{\mathbf{F}_\sigma = \sigma\kappa\boldsymbol{\nabla}C} \tag{1.148}$$

1.4.1.3 Smoothed interface representation

The smoothed interface representation, proposed by Sussman et al. [4] (1994) for a level set interface representation, writes the smeared out Heaviside function H_ε from the signed-distance function ϕ as presented earlier in this chapter. The expression is recalled here:

$$H_\varepsilon(\phi) = \begin{cases} 0 & \text{if } \phi < -\varepsilon \\ \frac{1}{2} + \frac{\phi}{2\varepsilon} + \frac{1}{2\pi}\sin\left(\frac{\pi\phi}{\varepsilon}\right) & \text{if } -\varepsilon \leq \phi \leq \varepsilon \\ 1 & \text{if } \phi > \varepsilon \end{cases} \tag{1.149}$$

where ε is half the thickness of the interface. The volumetric surface tension force then yields:

$$\boxed{\mathbf{F}_\sigma = \sigma\kappa\boldsymbol{\nabla}H_\varepsilon(\phi)} \tag{1.150}$$

1.4.1.4 Ghost-Fluid Method (GFM)

Using the Ghost-Fluid Method (GFM) of Fedkiw et al. (1999) [37], surface tension forces are embedded in the pressure gradient through the pressure jump at the interface $[p]_\Gamma$ defined by Eq. (1.29). This method for modeling surface tension is thus usually associated with the jump condition formulation (Section 1.1.3.3). However, it has been shown by [50] that it can be seen as a volumetric method as well, showing the equivalence between the two formulations.

Let us consider a one-dimensional numerical domain. A simple pressure gradient discretization within one of the phases at $i+1/2$ gives:

$$\boldsymbol{\nabla}p|_{i+1/2} = \frac{p_{i+1}-p_i}{\Delta x} \tag{1.151}$$

If the interface Γ is located in $[x_i;\, x_{i+1}]$, the pressure gradient accounts for the pressure jump $[p]_\Gamma$:

$$\boldsymbol{\nabla}p|_{\Gamma,i+1/2} = \frac{p_{i+1}-p_i}{\Delta x} - \frac{[p]_\Gamma}{\Delta x} = \boldsymbol{\nabla}p|_{i+1/2} - \frac{[p]_\Gamma}{\Delta x} \tag{1.152}$$

which yields for two fluids with equal dynamic viscosity coefficients:

$$\boldsymbol{\nabla}p|_{\Gamma,i+1/2} = \boldsymbol{\nabla}p|_{i+1/2} - \frac{(\sigma\kappa)|_{i+1/2}}{\Delta x} \tag{1.153}$$

The volumetric surface tension force is hence:

$$\mathbf{F}_\sigma|_{i+1/2} = (\sigma\kappa\delta)|_{i+1/2} \tag{1.154}$$

with

$$\delta|_{i+1/2} = \begin{cases} \pm 1/\Delta x & \text{if } \Gamma \in [x_i;\, x_{i+1}] \\ 0 & \text{otherwise} \end{cases} \tag{1.155}$$

The sign depends on the orientation of the interface relatively to the phases. The corresponding Heaviside function H is then:

$$H|_{i+1} = \begin{cases} 1 & \text{if } x_{i+1} \text{ is in liquid} \\ 0 & \text{if } x_{i+1} \text{ is in gas} \end{cases} \tag{1.156}$$

Indeed, if the interface Γ crosses the edge connecting the nodes i and $i+1$, with $i+1$ in liquid:

$$\mathbf{F}_\sigma|_{i+1/2} = (\sigma\kappa\delta)|_{i+1/2} = (\sigma\kappa\boldsymbol{\nabla}H)\,|_{i+1/2} = (\sigma\kappa)|_{i+1/2}\frac{H|_{i+1} - H|_i}{\Delta x} = \frac{(\sigma\kappa)|_{i+1/2}}{\Delta x} \tag{1.157}$$

The Ghost-Fluid Method can be used either with VOF interface-capturing schemes or with level set methods. Indeed, the condition "x_{i+1} in liquid" to compute the Heaviside function can be determined using either the condition $C > 0.5$ on the volume fraction, or $\phi > 0$ for distance level set schemes:

$$H(\mathbf{x}) = \begin{cases} 1 & \text{if } \phi(\mathbf{x}) > 0 \;\; (C > 0.5) \\ 0 & \text{otherwise} \end{cases} \tag{1.158}$$

To summarize, the GFM also belongs to volumetric formulations:

$$\boxed{\mathbf{F}_\sigma = \sigma\kappa\boldsymbol{\nabla}H(\mathbf{x})} \tag{1.159}$$

As explained in [50], a characteristic thickness of order Δx is also introduced in the GFM as in the other volumetric methods. Indeed, the approximation of the Heaviside function is insensitive to a shift of $\Delta x/2$ of the interface location.

The GFM is the surface tension modeling method used in this thesis. As a consequence, a more detailed presentation and derivation of the method, from the point of view of the jump condition formulation, and its implementation in YALES2 is given later in this manuscript (Section 2.3.2).

1.4.2 Well-balanced schemes - Discrete Laplace balance

If one considers the Navier-Stokes equations with surface tension for static fluids, one obtains the continuous balance between pressure and surface tension:

$$\boldsymbol{\nabla}p = \sigma\kappa\mathbf{n}\delta_A \tag{1.160}$$

which writes for volumetric formulations of surface tension:

$$\boxed{\boldsymbol{\nabla}p = \sigma\kappa\boldsymbol{\nabla}H} \tag{1.161}$$

Recover this balance numerically, even if not trivial to obtain, is crucial for capillary-driven flows, for which pressure and surface tension are often close to equilibrium. A procedure which allows to achieve discrete equilibrium is qualified as *well-balanced*. If this property is not verified, the numerical problem becomes quasi-static because of the arising of *spurious currents*, characterized by parasitic vortical velocity patterns around the liquid-gas interface (see Fig. 1.25).

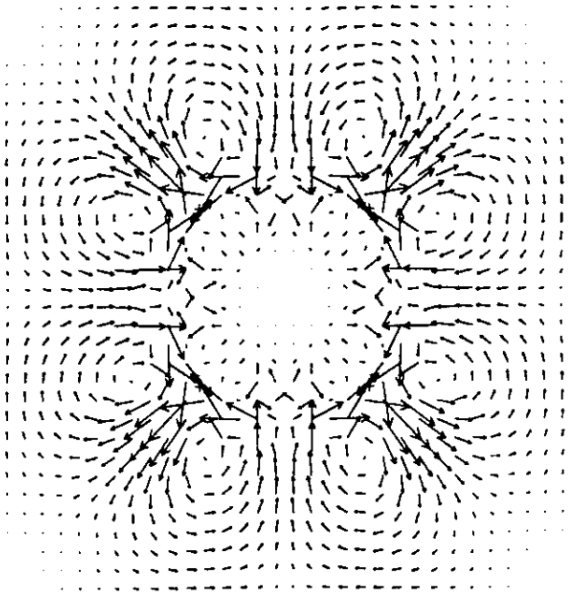


Figure 1.25: Spurious currents around a stationary bubble using a VOF method. Reprinted from [88].

A well-balanced procedure requires a common numerical operator to discretize the pressure gradient $\boldsymbol{\nabla} p$ and the Heaviside gradient $\boldsymbol{\nabla} H$. This condition leads to the exact cancellation of errors related to the discretization of these two terms. In this case, with a constant surface tension coefficient σ and a constant curvature κ, Eq. (1.161) writes [156, 50]:

$$p = \sigma \kappa H + \text{constant} \tag{1.162}$$

Thus, the well-known Laplace balance for a static droplet can be discretely recovered provided compatible gradient operators are applied to p and H, and that the curvature κ is constant [21].

However, the usual approximations of the Heaviside function H (volume fraction C for CSF, regularized Heaviside $H_\varepsilon(\phi)$, or $H(\mathbf{x})$ in the GFM) are sharp functions, and thus the estimation of $\boldsymbol{\nabla} H$ is very likely to be inaccurate. As a consequence, if the curvature computation derives from this gradient estimation through the interface normal $\mathbf{n}$ (e.g. $\kappa = \boldsymbol{\nabla} \cdot \mathbf{n} = \boldsymbol{\nabla} \cdot (\boldsymbol{\nabla} C / |\boldsymbol{\nabla} C|)$ for the CSF method), the resulting curvature will be even more inaccurate. Conversely, if one finds a better curvature discretization related to $\boldsymbol{\nabla} H$, one has to take care not to break the well-balanced property by introducing a different numerical operator than the one used for $\boldsymbol{\nabla} p$. It is thus crucial to decouple the evaluation of $\boldsymbol{\nabla} H$, which is necessary to balance $\boldsymbol{\nabla} p$ to yield a well-balanced scheme, from the curvature computation, which needs to be as accurate as possible. As explained in [50], this decoupling is naturally done in the level set method and its variants, as H is usually computed with the Ghost-Fluid Method (GFM), and κ is evaluated from the signed-distance function ϕ (either transported in SLS methods, or reconstructed in CLS methods), which is a smooth function.

In particular, in the ACLS framework used in this thesis, the GFM is employed for modeling surface tension and the signed-distance function ϕ is reconstructed geometrically to compute interface normals and curvature. As a consequence, as it will be seen in later, the curvature is calculated by differentiating a smooth function (Section 2.2.4.1), and the numerical operators used for this purpose are different from the ones employed for the treatment of surface tension within the pressure gradient, in the Poisson equation (Section 2.3.2.2, Eq. (2.57)). Moreover, common

differential operators are used for discretizing the pressure gradient without embedded surface tension $\boldsymbol{\nabla} p$ and the pressure jump $[p]_\Gamma \boldsymbol{\nabla} H = \sigma \kappa \boldsymbol{\nabla} H$ (Eqs. (2.57) and (2.59)).

In the next section, the main methods used in the literature to compute the interface curvature are presented, and the ability of the methods to obtain a constant curvature field at the interface, required to obtain the well-balanced property, is discussed.

1.4.3 Interface curvature computation

There are two main approaches in computing the interface curvature κ in interface-capturing methods: (i) by using the implicit representation of the interface $F(x, y, z) = 0$; (ii) by using an explicit representation of the interface $z = f(x, y)$, reconstructed from its implicit representation.

1.4.3.1 Implicit surface representation

For Level Set methods, the most natural way to compute interface normal $\mathbf{n}$ and curvature κ is to directly discretize the formulas given by differential geometry. Indeed, using the smooth signed-distance function ϕ yields accurate computations of normal and curvature. The interface normal writes:

$$\mathbf{n} = \frac{\boldsymbol{\nabla} \phi}{|\boldsymbol{\nabla} \phi|} \tag{1.163}$$

And the curvature is simply:

$$\boxed{\kappa = \boldsymbol{\nabla} \cdot \mathbf{n} = \boldsymbol{\nabla} \cdot \left(\frac{\boldsymbol{\nabla} \phi}{|\boldsymbol{\nabla} \phi|} \right)} \tag{1.164}$$

According to Goldman (2005) [157], the curvature can also be equivalently written as:

$$\boxed{\kappa = \frac{\mathrm{Tr}\left(\boldsymbol{\mathcal{H}}(\phi)\right) - \frac{\boldsymbol{\nabla}\phi^T}{|\boldsymbol{\nabla}\phi|} \cdot \boldsymbol{\mathcal{H}}\left(\phi\right) \cdot \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}}{|\boldsymbol{\nabla}\phi|}} \tag{1.165}$$

where $\boldsymbol{\mathcal{H}}(\phi) = \boldsymbol{\nabla}\boldsymbol{\nabla}\phi$ is the Hessian matrix of the signed-distance function. This last formula is used in this work, as it has proven to be more accurate than Eq. (1.164) in our numerical framework. The detailed discretization is given in the subsequent chapter. It should be noted that in this work, as the ACLS method is used, the signed-distance function ϕ is not transported and thus needs to be reconstructed from the interface location. This reconstruction is geometric from the explicit interface and is detailed in the next chapter.

Using Eqs. (1.164) or (1.165) yields a local computation of the curvature. It is thus necessary to then interpolate the local values to obtain the value at the interface. As mentioned in [50], this is a drawback when considering well-balanced properties and verifying discrete Laplace equilibrium at arbitrary resolutions, as it leads to a non-constant curvature field at the interface.

1.4.3.2 Explicit surface representation

A popular technique to compute the interface curvature is the height-function (HF) method [158, 159, 160]. It has been used especially in the framework of VOF methods, but implementations for level set methods have also been proposed. The HF algorithms rely on the definition of a local coordinate system in which the interface (line or plane for VOF-PLIC) is described explicitly as a graph of a function. In 2D, this leads to height and width function: $y = h_y(x)$ and $x = g_x(y)$, respectively. The choice of either of these functions depends on the local orientation of the interface normal $\mathbf{n}$. If $|n_y| > |n_x|$, $y = h_y(x)$ will be used, otherwise if $|n_x| > |n_y|$, $x = g_x(y)$

will be used [49]. The interface curvature κ is then obtained by differenciating the appropriate function:

$$\boxed{\kappa = \frac{h_y''(x)}{\left(1 + h_y'^2(x)\right)^{3/2}} \quad \text{or} \quad \kappa = \frac{g_x''(y)}{\left(1 + g_x'^2(y)\right)^{3/2}}} \tag{1.166}$$

Discretely, on a 2D cartesian grid, the height h and width g functions are constructed by summing the volume fractions C column-wise or row-wise, respectively:

$$h_i = \sum_j \Delta y C_{i,j} \quad \text{or} \quad g_j = \sum_i \Delta x C_{i,j} \tag{1.167}$$

where the number of cells taken into account for the sums is determined by the size of stencil used. According to Popinet (2009, 2018) [21, 50], from 3 to 9 cells may be required, depending on the interface configuration. For instance, using (3x7) and (7x3) stencils as in early implementations for height and width, respectively, leads to:

$$h_i = \sum_{k=-3}^{k=3} \Delta y C_{i,j+k} \quad \text{or} \quad g_j = \sum_{k=-3}^{k=3} \Delta x C_{i+k,j} \tag{1.168}$$

The discrete first and second derivatives are then:

$$\begin{cases} h_i' = \frac{h_{i+1} - h_{i-1}}{2\Delta x} \\ \\ h_i'' = \frac{h_{i+1} - 2h_i + h_{i-1}}{\Delta x^2} \end{cases} \tag{1.169}$$

or

$$\begin{cases} g_j' = \frac{g_{j+1} - g_{j-1}}{2\Delta y} \\ \\ g_j'' = \frac{g_{j+1} - 2g_j + g_{j-1}}{\Delta y^2} \end{cases} \tag{1.170}$$

eventually leading to the interface curvature estimation at cell (i, j) with Eq. (1.166). Situations exist where the height or width is not within the cell boundary. In this case, it is more accurate to interpolate the neighboring curvatures to obtain the local value. In should be noted that apart from this particular case, no further interpolation is needed: the curvature computed is directly obtained at the interface, which is a great advantage to obtain a constant curvature in the context of well-balanced schemes.

A weakness of HF methods is however known for under-resolved interfaces: consistent interface heights and widths become increasingly difficult to form as the interface resolution decreases, typically when $\kappa\Delta x < 1/5$ [50]. This can lead to failure of the curvature estimation using both horizontal and vertical procedures, because of the impossibility to differenciate twice the interface positions. Therefore, a generalized HF (GHF) curvature estimation has been proposed by Popinet (2009) [21], to perform a switch between the standard HF curvature estimation, used when the radius of curvature is resolved enough, and parabola-fitting methods with least-squares minimization such as the Parabolic Reconstruction Of Surface Tension (PROST) method of Renardy and Renardy (2002) [156], to increase the robustness of the overall procedure when the number of interface positions obtained is not sufficient to differenciate.

Recently, Karnakov et al. (2020) [161] proposed a more accurate method than parabolic fitting for increasing the robustness of the HF method in the case of under-resolved interfaces. In their method, particles are employed for estimating the curvature from the PLIC-reconstructed interface

(lines/planes). As with the GHF curvature computation of Popinet (2009) [21], this hybrid method provides a second-order convergent curvature on cartesian grids at high resolutions (standard HF), while reducing the errors at low interface resolutions using the particle method instead of parabolic fitting.

1.4.4 Explicit time integration - Capillary time-step restriction

When simulating interfacial liquid-gas flows using time-explicit methods, the temporal integration needs to be fine enough to capture the fastest capillary wave that can be captured by the mesh. A stability criterion related to capillary phenomena should then be added to the standard CFL criterion for the time-explicit interface transport.

A capillary wave propagating at the interface between a liquid and a gas follows the dispersion relation [162, 48]:

$$\omega_\sigma^2 = \frac{\sigma k^3}{\rho_l + \rho_g} \tag{1.171}$$

where ω_σ is the angular frequency, k is the wavenumber, ρ_l and ρ_g are liquid and gas densities, respectively. The phase velocity is defined as:

$$c_\sigma = \frac{\omega_\sigma}{k} = \sqrt{\frac{\sigma k}{\rho_l + \rho_g}} \tag{1.172}$$

Hence, the fastest capillary waves are those with the largest wavenumber $k_{\max}$, or minimum wavelength $\lambda_{\min} = 2\pi/k_{\max}$. The shortest wavelength resolved by the computational mesh, associated with the fastest capillary wave, is [48]:

$$\lambda_{\min} = 2\Delta x \tag{1.173}$$

with Δx the cell size. Therefore, the associated largest wavenumber is:

$$k_{\max} = \frac{2\pi}{\lambda_{\min}} = \frac{\pi}{\Delta x} \tag{1.174}$$

Thus, the phase velocity of the fastest capillary wave captured by the grid is:

$$c_{\sigma,\max} = \frac{\omega_\sigma}{k_{\max}} = \sqrt{\frac{\sigma k_{\max}}{\rho_l + \rho_g}} = \sqrt{\frac{\sigma\pi}{\Delta x(\rho_l + \rho_g)}} \tag{1.175}$$

Eventually, the corresponding minimum time-step according to Brackbill et al. [155] to temporally resolve the fastest capillary wave is:

$$\boxed{\Delta t < \Delta t_{\sigma,\min} = \frac{\Delta x}{2c_{\sigma,\max}} = \sqrt{\frac{\Delta x^3(\rho_l + \rho_g)}{4\pi\sigma}}} \tag{1.176}$$

The assumptions that lead to the 4π term are debatable [50, 163, 164]. For example, Denner and van Wachem (2015) presented a revised capillary timestep constraint in [164], taking into account the Doppler shift induced by the interface motion:

$$\Delta t < \Delta t_{\sigma,\min} = \frac{\Delta x}{\sqrt{2}c_{\sigma,\max} + \mathbf{u}_\Gamma \cdot \mathbf{k}/|\mathbf{k}|} \tag{1.177}$$

where $\mathbf{u}_\Gamma$ is the velocity at the interface and $\mathbf{k}$ the wavenumber vector.

The capillary time-step restriction leads to strong constraints on the time step, especially when refining the grid: Δt scales as $\Delta x^{3/2}$ and thus this constraint becomes dominant over the transport CFL restriction at fine mesh resolutions. Nevertheless, as widely done in the literature, a time-explicit discretization of surface tension is used in this work, and the capillary time-step constraint used in YALES2 is presented in the subsequent chapter. Implicit time-stepping methods remain an active research topic: alleviate the strong constraint on the time step could indeed be particularly helpful for simulating small-scale flows, such as in microfluidic applications. A review of existing implicit time-stepping methods can be found in [50].

CHAPTER 2

A two-phase flow solver for adaptive unstructured grids

Disclaimer: Part of this chapter has been published in *Journal of Computational Physics* [216].

Contents

2.1 Overview of the framework 56
2.1.1 Accurate Conservative Level Set 56
2.1.2 Navier-Stokes equations 58
2.1.3 Summary of the full solution procedure 58
2.2 Accurate Conservative Level Set algorithm 59
2.2.1 Narrow-band ACLS 59
2.2.2 Signed-distance function reconstruction: the GPMM algorithm 59
2.2.3 Finite-volume framework for unstructured grids: spatial discretization and notations 64
2.2.4 Interface normals and curvature computation 65
2.2.5 Computation of the interface profile thickness 66
2.2.6 Discretization of the reinitialization equation 66
2.2.7 Transport of the ACLS function - Time integration 67
2.3 Incompressible flow solver 69
2.3.1 Velocity prediction with momentum correction 69
2.3.2 Ghost-Fluid Method 70
2.3.3 Time integration - Capillary timestep restriction 72
2.4 Isotropic dynamic mesh adaptation strategy 72
2.4.1 Target metric generation 73
2.4.2 Automatic triggering 73
2.4.3 Parallel iterative remeshing 74

This second chapter is dedicated to the presentation of the YALES2 two-phase liquid-gas flow solver. A brief overview of the mathematical framework and the full procedure are first given. Then, the detailed numerical implementation of the Accurate Conservative Level Set (ACLS) interface-capturing algorithm for unstructured grids (triangular and tetrahedral) is presented, with an emphasis on the geometric reconstruction of the signed distance to the interface, and on the discretized reinitialization equation. The consistent coupling of the interface transport with the incompressible flow solver is then detailed, and surface tension forces, treated using the Ghost-Fluid Method (GFM), are integrated in the projection method through the pressure jump, itself within the pressure gradient. The time-integration being explicit, these surface tension forces constrain the time step, to be able to resolve the fastest capillary waves in the domain. Eventually, the isotropic Adaptive Mesh Refinement (AMR) strategy, employed to concentrate the mesh cells around the liquid-gas interface, is presented.

2.1 Overview of the framework

2.1.1 Accurate Conservative Level Set

In the Accurate Conservative Level Set (ACLS) framework [31], the liquid-gas interface Γ is represented using a hyperbolic tangent profile:

$$\psi(\mathbf{x},t) = \frac{1}{2}\left(\tanh\left(\frac{\phi(\mathbf{x},t)}{2\varepsilon}\right)+1\right) \tag{2.1}$$

where the parameter ε sets the thickness of the profile, and $\phi(\mathbf{x},t) = \pm|\mathbf{x}(t)-\mathbf{x}_\Gamma(t)|$ is the signed-distance function.
Using ψ, the interface is located at the iso-level $1/2$:

$$\Gamma(t) = \{\mathbf{x}\in\mathbb{R}^3 |\, \psi(\mathbf{x},t) = 1/2\} \tag{2.2}$$

as depicted in the one-dimensional example of Fig. 2.1. In this figure, a characteristic profile

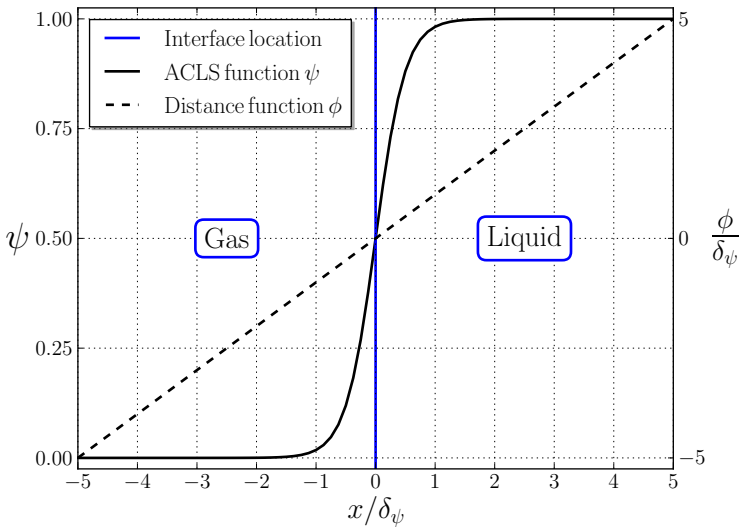


Figure 2.1: Interface capturing on a one-dimensional example using the ACLS method. $x_\Gamma = 0$; $\delta_\psi = 4\varepsilon$.

thickness δ_ψ is used, computed based on the maximum gradient of ψ:

$$\delta_\psi = \frac{1}{\max|\boldsymbol{\nabla}\psi|} = 4\varepsilon \tag{2.3}$$

Assuming the flow velocity field $\mathbf{u}$ is divergence free, the scalar ψ is advected via the following equation:

$$\frac{\partial\psi}{\partial t} + \boldsymbol{\nabla}\cdot(\psi\mathbf{u}) = 0 \tag{2.4}$$

and then reshaped, at every time step, using the reinitialization equation of [31]:

$$\frac{\partial\psi}{\partial\tau} = \boldsymbol{\nabla}\cdot\Big(\underbrace{\varepsilon(\boldsymbol{\nabla}\psi\cdot\mathbf{n})\mathbf{n}}_{\text{Diffusion}} - \underbrace{\psi(1-\psi)\mathbf{n}}_{\text{Resharpening}}\Big) \tag{2.5}$$

where τ is a pseudo-time, and $\mathbf{n}$ is the interface normal:

$$\mathbf{n} = \frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|} \tag{2.6}$$

A new form of the reinitialization equation was recently proposed by Chiodi et al. in [35]:

$$\frac{\partial\psi}{\partial\tau} = \boldsymbol{\nabla}\cdot\left(\frac{1}{4\cosh^2\left(\frac{\phi_{\text{map}}}{2\varepsilon}\right)}\left(|\boldsymbol{\nabla}\phi_{\text{map}}\cdot\mathbf{n}|-1\right)\mathbf{n}\right) \tag{2.7}$$

where $\phi_{\text{map}} = \varepsilon\ln\left(\psi/(1-\psi)\right)$ is an analytical signed-distance function, mapped for $\psi\in]0;1[$. More details regarding the history of conservative level-set reinitialization equations can be found in Section 1.3.2.2. As detailed in [31] and [35], Eqs. (2.5) and (2.7) converge fast and thus solving one or the other in a fixed small number of pseudo-time steps $\Delta\tau$ with $\Delta\tau\sim\Delta x\sim\varepsilon$ is sufficient to reach steady state. In the present work, for the sake of simplicity, 3 steps are performed at every iteration instead of checking the convergence of Eq. (2.5) or (2.7).

The signed-distance function ϕ is reconstructed at nodes in the narrow band around the interface using a Geometric-Projection Marker Method (GPMM), firstly introduced in [165], to estimate the smallest distance to the interface. The mean curvature κ is computed directly from ϕ using Goldman's formula [157]:

$$\kappa = \frac{\text{Tr}\left(\boldsymbol{\mathcal{H}}(\phi)\right) - \frac{\boldsymbol{\nabla}\phi^T}{|\boldsymbol{\nabla}\phi|}\cdot\boldsymbol{\mathcal{H}}\left(\phi\right)\cdot\frac{\boldsymbol{\nabla}\phi}{|\boldsymbol{\nabla}\phi|}}{|\boldsymbol{\nabla}\phi|} \tag{2.8}$$

where $\boldsymbol{\mathcal{H}}(\phi)$ is the Hessian matrix of the signed-distance function.

The ACLS method is not strictly conservative for two reasons. First, the approximate and exact enclosed volumes of liquid are respectively:

$$\widetilde{V}_l = \int_V \psi\,\mathrm{d}V \quad\text{and}\quad V_l = \int_V H(\psi-1/2)\,\mathrm{d}V \tag{2.9}$$

where H is the Heaviside function (0 in gas, 1 in liquid). In the ACLS method, the approximate volume $\widetilde{V}_l$ is conserved, but differs from the exact volume V_l due to interface curvature, i.e. in the ubiquitous case of non-straight interfaces.

A second reason for losing liquid mass on irregular meshes is the variation of the mesh resolution from fine to coarse, which leads to the smoothing of ψ, and may imply a loss of the $\psi=1/2$ iso-surface. In this case, the interface is not detected anymore and gas can be set instead of liquid.

Finally, the reinitialization process on non-homogeneous meshes may also induce errors in the transport speed: the profile thickness parameter ε varies based on the local grid resolution, typically set to $\Delta x/2$ (more details about the computation of ε are provided in Section 2.2.5). Spatial variations of this parameter induce changes in the profile thickness, which in turn may induce errors on the transport velocity of the $\psi=1/2$ iso-surface.

Because of all these numerical errors on non-homogeneous meshes, special attention is needed when coupling the ACLS procedure with dynamic mesh adaptation, so that the interface never encounters strong cell-size gradients.

In this work, two types of cell size Δx will be used to characterize unstructured grid discrete elements: (i) a node-based cell size Δx_i, which characterizes the local mesh resolution at node i, computed from the node-based control volume V_i (see Fig. 2.5): $\Delta x_i = \sqrt{V_i}$ in 2D and $\Delta x_i = \sqrt[3]{V_i}$ in 3D; (ii) the length of the pair of nodes Δx_{ik}, which will be used for the pair-based numerical schemes presented later in this manuscript.

2.1.2 Navier-Stokes equations

The momentum equation for an incompressible Newtonian fluid is written in each of the phases as:

$$\frac{\partial \mathbf{u}}{\partial t} + \boldsymbol{\nabla} \cdot (\mathbf{u} \otimes \mathbf{u}) = -\frac{1}{\rho}\boldsymbol{\nabla} p + \boldsymbol{\nabla} \cdot \boldsymbol{\tau} + \mathbf{g} \tag{2.10}$$

with $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$. As presented in Section 1.1.3.3, the flow velocity is assumed continuous in both normal and tangential directions across the liquid/gas interface Γ, and hence there is no velocity jump: $[\mathbf{u}]_\Gamma = 0$. However, pressure and density are discontinuous across Γ [53]. Surface tension σ is embedded in the pressure jump $[p]_\Gamma$, and, without Marangoni forces, the pressure jump is written as:

$$[p]_\Gamma = p_{l,\Gamma} - p_{g,\Gamma} = \sigma\kappa_\Gamma + 2[\mu]_\Gamma \mathbf{n}^t \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{n} \tag{2.11}$$

Numerically, it is treated explicitly using the Ghost-Fluid Method (GFM) [37]. In Eq. (2.11), $\mathbf{n}$ is the interface normal from Eq. (2.6), κ_Γ is the interface mean curvature, $p_{l,\Gamma}$ and $p_{g,\Gamma}$ are the liquid and gas pressure at the interface, respectively. The material properties are reconstructed as:

$$\rho(\mathbf{x},t) = \rho_g + (\rho_l - \rho_g)H(\psi(\mathbf{x},t) - 1/2) \text{ and } \mu(\mathbf{x},t) = \mu_g + (\mu_l - \mu_g)\psi(\mathbf{x},t) \tag{2.12}$$

for the density and the dynamic viscosity, respectively. As in the original CLS method of Olsson and Kreiss (2005) [29] and original ACLS method of Desjardins et al. (2008) [31], the smoothed Heaviside function ψ is used to reconstruct the viscosity field, increasing the robustness of the method: the Ghost-Fluid Method (GFM) is indeed challenging to apply in the presence of the viscous term [31]. Moreover, as mentioned in [31], for turbulent flows, the viscous terms are expected to be significant only at the smallest scales, which do not contain much energy. Hence, the choice of smoothing the viscosity discontinuity is not likely to influence significantly the quality of the solution.

A projection method [57], presented in Section 1.2.1, is employed to solve the incompressible Navier-Stokes equations with surface tension. The linear solver used to solve the Poisson equation for pressure is the Deflated Preconditioned Conjugate Gradient of [59].

2.1.3 Summary of the full solution procedure

To ease the reading of the present chapter, the full solution procedure is here summarized:

(i) Using the unstructured ACLS algorithm, advance the interface implicitly through the ψ field with Eq. (2.37) using $\mathbf{u}^n$ (Section 2.2.7), then reinitialize ψ to reform the hyperbolic tangent profile with Eq. (2.31) (Section 2.2.6).

(ii) Set the new narrow band around the interface (Section 2.2.1).

(iii) Compute interface features in the narrow band: geometric signed distance ϕ_{GPMM} (Section 2.2.2), normals $\mathbf{n}$ (Eq. (2.25); Section 2.2.4), curvature κ_Γ (Eq. (2.26) then Eq. (2.29); Section 2.2.4).

(iv) Predict the velocity field $\mathbf{u}^*$ from $\mathbf{u}^n$, using the momentum correction at the interface of [166] adapted to unstructured meshes (Section 2.3.1).

(v) Project the velocity field by solving the Poisson equation, in which interface jump conditions are treated with the Ghost-Fluid Method (Eq. (2.57); Section 2.3.2).

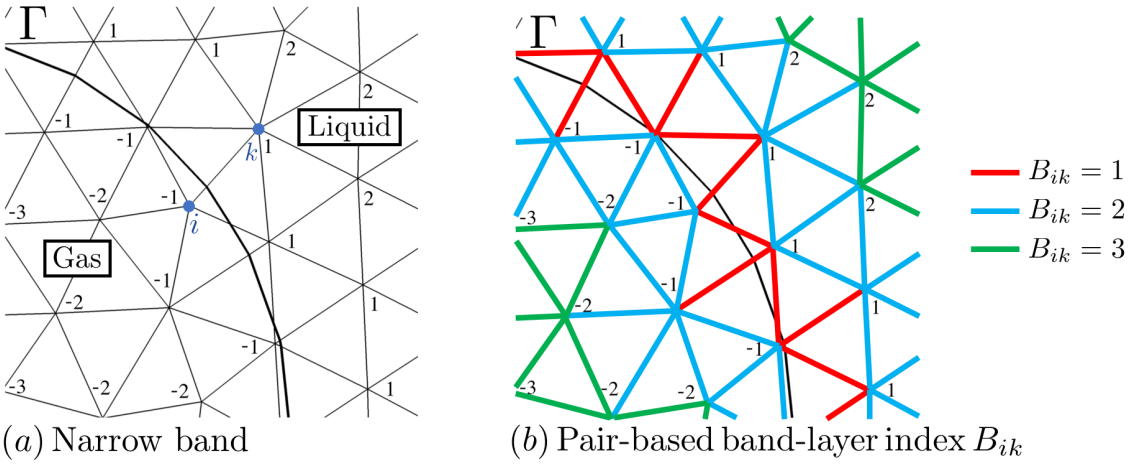
(vi) Correct the velocity, with the pressure gradient that includes the jump conditions using the Ghost-Fluid Method (Eq. (2.58); Section 2.3.2), eventually leading to $\mathbf{u}^{n+1}$ so that $\boldsymbol{\nabla} \cdot \mathbf{u}^{n+1} = 0$.

(vii) Grid adaptation (Section 2.4).

2.2 Accurate Conservative Level Set algorithm

2.2.1 Narrow-band ACLS

In order to reduce computational costs, interface features such as signed distance, normals, curvature and reinitialization fluxes are computed in a restricted region around the interface, called "narrow band" [36]. Each node i belonging to the narrow band is identified with a signed integer b_i ($b > 0$ in liquid and $b < 0$ in gas), depending on the node layer it is part of. The flagging of the narrow band is illustrated in Fig. 2.2 (a). Using this methodology, an edge ik crossing the interface



(a) Narrow band (b) Pair-based band-layer index B_{ik}

Figure 2.2: (a) Flagging of the narrow band around the interface Γ, here ik is a pair of nodes crossing the interface; (b) Pair-based band-layer index B_{ik} for data propagation in the Geometric-Projection Marker Method (GPMM).

verifies $b_i b_k = -1$. In practice, the band width is set between 8 and 12 cells on each side of the interface: $8 \leq |b|_{\max} \leq 12$.

2.2.2 Signed-distance function reconstruction: the GPMM algorithm

To compute the interface normals (Eq. (2.6)) and curvature (Eq. (2.8)), a reconstructed signed-distance function ϕ is used. As detailed in [31], using this smooth function instead of ψ for the computation of normal avoids spurious orientation of this normal vector, which is thus appropriate for the reinitialization equation. This distance reconstruction is challenging on unstructured meshes, and very few massively parallel algorithms are able to do it. In the work of Dapogny and Frey [167], the signed-distance function is reconstructed on unstructured meshes from a triangulation of the interface using the unsteady Eikonal equation. In particular, the reduction of triangulation-induced numerical errors when refining the mesh close to the interface is discussed and quantified. In this work, the interface is also approximated by a triangulation before reconstructing the distance, but it has been chosen to proceed geometrically using a pair-based Fast-

Marching Method (FMM) for its computation from the explicit surface. The methodology is the following:

(i) A preliminary sorting of the node pairs in the narrow band is performed to be able to sweep from the close-interface band layers to the ones further;

(ii) A segmentation (2D) or triangulation (3D) of the interface within grid cells is done;

(iii) A Fast-Marching Method (FMM) is then performed along the edges to build a collection of the closest segments (2D) or triangles (3D) at each node of the band;

(iv) The signed-distance function is computed from an exact geometric projection at each node of the band.

All these steps constitute the Geometric-Projection Marker Method (GPMM), first introduced in [165] and presented here in more details.

2.2.2.1 Sorting of the node pairs

In order to propagate data efficiently from the first band level $|b| = 1$ to the furthest $|b| = |b|_{\max}$, sorting the pairs of nodes layer by layer in the narrow band is necessary. To do so, a pair-based band-layer index B_{ik} is defined as follows:

$$B_{ik} = \begin{cases} 1 \text{ if } b_i b_k = -1 \\ \min\left(|b_i|, |b_k|\right) + 1 \text{ if } b_i b_k \neq -1 \end{cases} \tag{2.13}$$

with $1 \leq B_{ik} \leq |b|_{\max} + 1$. This index, depicted in Fig. 2.2 (b), is computed by looping on the pairs ik and using pair-to-node connectivity.

2.2.2.2 Determination of intersection points

It is then needed to compute the position $\mathbf{x}_P$ of the intersection points P between the interface Γ and the edges ik crossing the interface ($B_{ik} = 1$). Given the conservative level-set functions ψ_i and ψ_k (Eq. (2.1)) at the nodes i and k belonging to a crossing pair ik, one can invert these functions to build two distance functions to the interface $\phi_{\text{inv},i}$ and $\phi_{\text{inv},k}$:

$$\begin{cases} \phi_{\text{inv},i} = 2\varepsilon_i \text{atanh}\left(2\psi_i - 1\right) \\ \phi_{\text{inv},k} = 2\varepsilon_k \text{atanh}\left(2\psi_k - 1\right) \end{cases} \tag{2.14}$$

The level-set position θ on a pair ik crossing the interface is then calculated as:

$$\theta = \frac{\phi_{\text{inv},i}}{\phi_{\text{inv},i} - \phi_{\text{inv},k}} \tag{2.15}$$

which means that $\theta = 0$ and $\theta = 1$ correspond to Γ at nodes i and k, respectively. Thus, if the pair of nodes ik verifies the condition $(\psi_i - 0.5)(\psi_k - 0.5) < 0$, the position of the intersection point is eventually computed by interpolating the nodal coordinates as:

$$\mathbf{x}_P = \mathbf{x}_i + \theta(\mathbf{x}_k - \mathbf{x}_i) = \mathbf{x}_i + \theta \boldsymbol{\Delta x}_{ik} \tag{2.16}$$

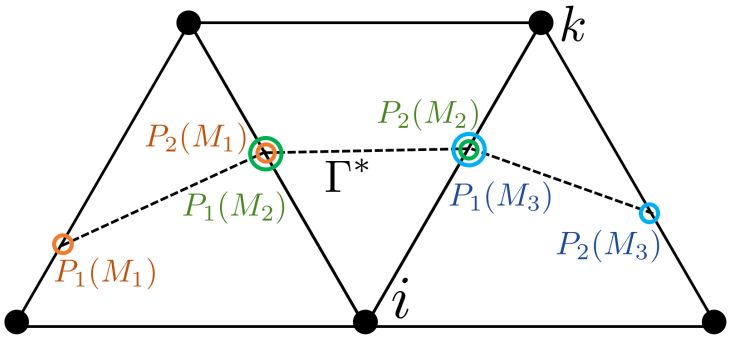


Figure 2.3: Segmentation of the interface in 2D. $P_K(M_I)$ is the K-th intersection point associated with the I-th marker M_I.

2.2.2.3 Segmentation/Triangulation of the interface

In order to compute the distance to an explicit interface, noted Γ^* in the following, a segmentation in 2D or a triangulation in 3D of the implicit interface Γ is first performed. This step is illustrated in 2D in Fig. 2.3. The concept of marker stems from this procedure. A marker M is an array containing the coordinates $\mathbf{x}_K(M) = (x_K(M), y_K(M), z_K(M))$ of its associated intersection points $P_K(M)$ seen in Fig. 2.3. Hence, for 2D triangular grids, each interface segment is defined by a set of two intersection points ($K = 1, 2$) stored in a marker. For 3D tetrahedral meshes, there can be three or four intersection points depending on how the interface intersects a tetrahedron. For a three-point intersection, the interface within the cell is defined by a single triangle. In the case of a four-point intersection, the resulting quadrilateral is divided into four triangles to ease the storing of the surface elements, leading in both cases to three points stored in one marker in 3D ($K = 1, ..., 3$).

Each node of the narrow band knows a maximum fixed number of markers N_M defined by the user ($I = 1, ..., N_M$), which is usually set to 3 in 2D and 10 in 3D. Indeed, as studied in [165], propagating several markers significantly improves the accuracy and spatial convergence of the method. A set of markers known by a node i of the narrow band is a marker list. Each marker M, which is a node-based array M_i, also contains the absolute distance $d(M_i)$ from the node i to the associated surface element, and two identificators $id_{1,2}(M_i)$. id_1 is the processor number and id_2 is the mesh element number in which the intersection points are. A 2D nodal marker list is presented in Table 2.1.

Table 2.1: A 2D marker list for a node i of the narrow band: 2 identificators $id_{1,2}$, 4 coordinates ($K = 1, 2$), 1 distance d.

$M_{1,i}$	...	$M_{I,i}$	...	$M_{N_M,i}$
$id_1(M_{1,i})$	...	$id_1(M_{I,i})$	...	$id_1(M_{N_M,i})$
$id_2(M_{1,i})$	...	$id_2(M_{I,i})$	...	$id_2(M_{N_M,i})$
$x_1(M_{1,i})$	...	$x_1(M_{I,i})$	...	$x_1(M_{N_M,i})$
$y_1(M_{1,i})$	...	$y_1(M_{I,i})$	...	$y_1(M_{N_M,i})$
$x_2(M_{1,i})$	...	$x_2(M_{I,i})$	...	$x_2(M_{N_M,i})$
$y_2(M_{1,i})$	...	$y_2(M_{I,i})$	...	$y_2(M_{N_M,i})$
$d(M_{1,i})$	...	$d(M_{I,i})$	...	$d(M_{N_M,i})$

The markers will then be sorted by distance for each node i, within the marker list so that:

$$d(M_{1,i}) < ... < d(M_{I,i}) < ... < d(M_{N_M,i}) \tag{2.17}$$

2.2.2.4 Sorting of markers and distance computation

When a marker is evaluated to update and fill marker lists, a check is first performed based on the two marker identificators to identify whether the marker is already present in the marker list or not. If not, the marker distance needs to be estimated to place the new marker in the list. Two cases are then considered based on the nodal band-layer index b_i, and a threshold $|b|_{\text{exact}} = 2$: (i) in the first two band levels, i.e. if $|b_i| \leq |b|_{\text{exact}}$, the distance d is computed by performing an exact geometric projection; (ii) for the furthest band levels, i.e. if $|b|_{\text{exact}} < |b_i| \leq |b|_{\max}$, the distance is initially approximated in order to speed-up the marker-sorting procedure. We focus on 2D triangular grids in the following for sake of simplicity.

In case (i), the geometric projection algorithm is used, illustrated in Fig. 2.4. If the projection

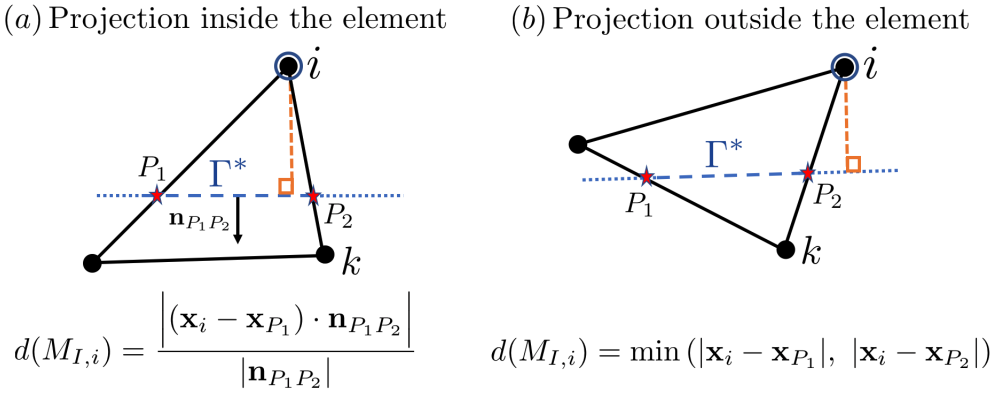


Figure 2.4: Projection algorithm to compute the exact geometric absolute distance d to the triangulated interface Γ^*.

is inside the element (Fig. 2.4 (a)), the distance to the segment $[P_1P_2]$ stored in marker $M_{I,i}$ at node i is computed geometrically as:

$$d(M_{I,i}) = \frac{\left|(\mathbf{x}_i - \mathbf{x}_{P_1}) \cdot \mathbf{n}_{P_1P_2}\right|}{\left|\mathbf{n}_{P_1P_2}\right|} \tag{2.18}$$

where $\mathbf{n}_{P_1P_2}$ is the normal vector to the interface segment $[P_1P_2]$. Otherwise, if the projection is outside the element (Fig. 2.4 (b)), the distance d is the distance between the node i and the closest intersection point P_K:

$$d(M_{I,i}) = \min(|\mathbf{x}_i - \mathbf{x}_{P_1}|,\ |\mathbf{x}_i - \mathbf{x}_{P_2}|) \tag{2.19}$$

For case (ii), the distance is simply approximated for the sorting as:

$$d(M_{I,i}) = \min\left(\left|\mathbf{x}_i - \mathbf{x}_{P_1}\right|,\ \left|\mathbf{x}_i - \mathbf{x}_{P_2}\right|,\ \frac{\left|(\mathbf{x}_i - \mathbf{x}_{P_1}) + (\mathbf{x}_i - \mathbf{x}_{P_2})\right|}{2}\right) \tag{2.20}$$

Once the marker distance is estimated, it is then needed to sort the marker by distance and thus include the new one in the nodal marker list. This procedure is presented in Algorithm 1.

Algorithm 1 Update of a marker list at a node i of the narrow band. The new marker to be included in the list is $M_{*,i}$.

1: **if** $d(M_{*,i}) \leq d(M_{1,i})$ **then**
2: **for** $I = N_M - 1, 1, -1$ **do**
3: $M_{I+1,i} = M_{I,i}$
4: **end for**
5: $M_{1,i} = M_{*,i}$
6: **else if** $d(M_{*,i}) \geq d(M_{N_M,i})$ **then**
7: **return**
8: **else**
9: **for** $I = 1, N_M - 1$ **do**
10: **if** $d(M_{*,i}) < d(M_{I+1,i})$ **then**
11: **for** $I' = N_M - 1, 1 + I, -1$ **do**
12: $M_{I'+1,i} = M_{I',i}$
13: **end for**
14: $M_{I+1,i} = M_{*,i}$
15: **return**
16: **end if**
17: **end for**
18: **end if**

2.2.2.5 Optimization and parallelization

The present algorithm is iterative for parallelism purposes. YALES2 parallelization uses a double-domain decomposition: within a processor, grid elements are assembled in groups. To optimize the algorithm, it is necessary to check local changes, within a cell group or a processor.

In addition to the marker list array, another array which contains marker flags $\mathcal{F}(M)$ (0 or 1) is created before propagating data (i.e. after treating the first band level) in the narrow band. $\mathcal{F}_i(M) = 1$ means that the marker M is new in the list at node i: e.g. this flag would be turned to 1 right after lines 5 and 14 of Algorithm 1. This operation allows to have updated information on the status of markers during data propagation, and thus avoids unnecessary repetition of computational operations (e.g. redundant updates and sorting of markers).

Another optimization is to store the squared distance $d^2(M_{I,i})$ in marker lists instead of the distance $d(M_{I,i})$, to avoid many calculations of square roots, which have seen to be very expensive due to the large number of distance computations in the present algorithm. Indeed, this minor change does not affect the comparison and sorting of markers.

2.2.2.6 Final nodal signed-distance computation in the narrow band

Once all markers are sorted, $M_{1,i}$ contains the smallest squared distance $d^2(M_{1,i})$. The signed-distance is thus simply computed at node i as $\phi_i = \pm\sqrt{d^2(M_{1,i})}$ if $|b_i| \leq |b|_{\text{exact}}$, otherwise for the nodes where the distance d in markers has been only approximated for the sorting ($|b|_{\text{exact}} < |b_i| \leq |b|_{\max}$), the distance ϕ_i is eventually computed using the projection algorithm presented above (Fig. 2.4) onto the closest surface element, known thanks to the marker sorting.

2.2.3 Finite-volume framework for unstructured grids: spatial discretization and notations

As seen in Chapter 1, YALES2 is a pair-based finite-volume incompressible solver with node-centered control volumes [38]. To ease the reading of the present chapter, the main elements and formulas used in the following sections are briefly recalled.

The control volumes V_i are based on the dual mesh, as shown in Fig. 2.5. The spatial discretization is ensured by 4th-order central finite-volume schemes, written on the dual mesh constructed from edge and element centers (dual cells connect centroids of cells to centroids of faces). In this

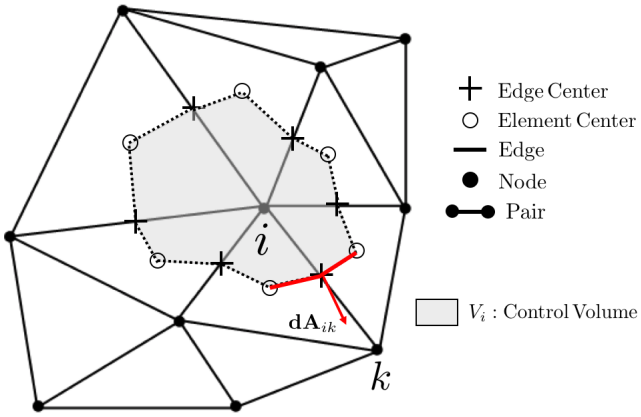


Figure 2.5: Finite-volume pair-based discretization in YALES2, using node-centered control volumes V_i.

framework, presented earlier in Section 1.2.4, second and fourth-order gradients of a quantity ζ are computed at a given node i using the gradient theorem as:

$$\boldsymbol{\nabla}\zeta|_i = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\zeta_{ik}\mathbf{dA}_{ik} \tag{2.21}$$

with $\mathcal{N}_i$ the set of neighbors of node i, ζ_{ik} the pair-based estimation of ζ, computed to either second or fourth order, and $\mathbf{dA}_{ik}$ the normal vector to the portion of the control volume V_i, associated to pair ik (red solid line in Fig. 2.5). Second and fourth-order approximations of the gradient are written as:

$$\boldsymbol{\nabla}\zeta|_i^{\text{2nd}} = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\left(\frac{\zeta_i+\zeta_k}{2}\right)\mathbf{dA}_{ik} \tag{2.22}$$

$$\boldsymbol{\nabla}\zeta|_i^{\text{4th}} = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\left(\frac{\zeta_i+\zeta_k}{2}+\frac{\boldsymbol{\nabla}\zeta|_i^{\text{2nd}}-\boldsymbol{\nabla}\zeta|_k^{\text{2nd}}}{6}\cdot\boldsymbol{\Delta x}_{ik}\right)\mathbf{dA}_{ik} \tag{2.23}$$

where $\boldsymbol{\Delta x}_{ik}$ is the direction of the pair ik. It is useful to define the unit pair direction $\mathbf{e}_{ik} = \boldsymbol{\Delta x}_{ik}/\Delta x_{ik}$, which will be used in the following sections. Using the divergence theorem, the divergence of a flux $\mathbf{F}$ writes:

$$\boldsymbol{\nabla}\cdot\mathbf{F}|_i = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\mathbf{F}_{ik}\cdot\mathbf{dA}_{ik} \tag{2.24}$$

with $\mathbf{F}_{ik}$ the pair-based flux computed as in the gradient to either second or fourth order.

2.2.4 Interface normals and curvature computation

The interface normals and curvature are computed in the narrow band around the interface, using the reconstructed signed-distance function ϕ from the GPMM algorithm presented above.

The interface normals are pair-based, and computed as:

$$\mathbf{n}_{ik} = \boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_{ik}^{4\mathrm{th}} = (\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_{i}^{4\mathrm{th}} + \boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_{k}^{4\mathrm{th}})/2 \tag{2.25}$$

where $\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_{i\,\mathrm{or}\,k}^{4\mathrm{th}}$ are fourth-order nodal gradients, computed using Eq. (2.23).

The interface curvature κ_Γ is computed in two steps: (i) the curvature is first estimated at nodes (κ_i) in the narrow band using Eq. (2.8); (ii) for the node pairs crossing the interface ($b_i b_k = -1$), a linear interpolation of the nodal curvature radii $R_i = 1/\kappa_i$ is performed to obtain κ_Γ.

2.2.4.1 Computation of curvature at nodes

The local mean curvature at a node i is given by discretizing Eq. (2.8):

$$\kappa_i = \frac{\mathrm{Tr}\left(\boldsymbol{\mathcal{H}}_i(\phi_{\mathrm{GPMM}})\right) - \frac{\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}^T|_i^{2\mathrm{nd}}}{\left|\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_i^{2\mathrm{nd}}\right|} \cdot \boldsymbol{\mathcal{H}}_i\left(\phi_{\mathrm{GPMM}}\right) \cdot \frac{\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_i^{2\mathrm{nd}}}{\left|\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_i^{2\mathrm{nd}}\right|}}{\left|\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_i^{2\mathrm{nd}}\right|} \tag{2.26}$$

where $\boldsymbol{\mathcal{H}}_i(\phi_{\mathrm{GPMM}})$ is the discretized Hessian operator at node i applied to the reconstructed signed-distance function ϕ_{GPMM}. Two discretizations of the Hessian operator are studied in this work: compact and non-compact. The compact Hessian computation necessitates only neighboring nodes (i.e. a compact stencil), and is written as:

$$\boldsymbol{\mathcal{H}}_i^{\mathrm{C}}(\phi_{\mathrm{GPMM}}) = \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\left[\underbrace{\frac{\phi_{\mathrm{GPMM},k} - \phi_{\mathrm{GPMM},i}}{\Delta x_{ik}}\mathbf{e}_{ik}}_{\text{Aligned with } ik} + \underbrace{\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_{ik}^{2\mathrm{nd}} - \left(\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|_{ik}^{2\mathrm{nd}} \cdot \mathbf{e}_{ik}\right)\mathbf{e}_{ik}}_{\text{Orthogonal to } ik}\right] \otimes \mathbf{dA}_{ik} \tag{2.27}$$

whereas the non-compact formulation is simply obtained by applying two consecutive second-order gradients, leading to a non-compact stencil:

$$\boldsymbol{\mathcal{H}}_i^{\mathrm{NC}}(\phi_{\mathrm{GPMM}}) = \boldsymbol{\nabla}\left(\boldsymbol{\nabla}\phi_{\mathrm{GPMM}}|^{2\mathrm{nd}}\right)\Big|_i^{2\mathrm{nd}} \tag{2.28}$$

2.2.4.2 Interpolation for crossing pairs

Once the nodal curvatures are computed, a linear interpolation of the curvature radii is performed for the pairs ik crossing the interface Γ, resulting in a pair-based interface curvature:

$$\kappa_{\Gamma,ik} = \frac{1}{\theta R_i + (1-\theta)R_k} = \frac{\kappa_i\kappa_k}{\theta\kappa_k + (1-\theta)\kappa_i} \tag{2.29}$$

with θ the interpolation weight ($0 < \theta < 1$), defined in Eq. (2.15), representing the position of the interface on the crossing pair ik.

2.2.5 Computation of the interface profile thickness

As mentioned in Section 2.1.1, the parameter ε, which sets the thickness of the hyperbolic tangent profile ψ, depends on the local mesh resolution: $\varepsilon = \varepsilon(\mathbf{x})$. This parameter is computed at nodes as follows:

$$\varepsilon_i = \varepsilon_F \frac{\sum_{k\in\mathcal{N}_i} \Delta x_{ik} |\mathbf{dA}_{ik}|}{\sum_{k\in\mathcal{N}_i} |\mathbf{dA}_{ik}|} \tag{2.30}$$

where $\mathcal{N}_i$ is the set of neighbors of node i, ε_F is a constant scaling factor: typically $1/2$, Δx_{ik} is the pair length, and $\mathbf{dA}_{ik}$ is the surface defined in Fig. 2.5. As presented in detail recently by Waclawczyk in [168], spatial variations of the interface profile thickness ε may cause a departure from equilibrium in the intermittency region, i.e. the transient part of the ψ profile. In this case, an extension of the mapping procedure used to obtain Eq. (2.7) is needed. However, in this work, by performing Adaptive Mesh Refinement (AMR) around the interface, the mesh is homogeneous around it and the variations of ε from one node to another are expected to be minor. Thus, in this manuscript, the intermittency region is assumed to be in the equilibrium state. This assumption does not impact neither the normal nor the curvature computation, as they are both estimated from the geometrically-reconstructed signed-distance function ϕ_{GPMM}: they do not depend analytically on the ψ-function.

2.2.6 Discretization of the reinitialization equation

The reformulated reinitialization (Eq. (2.7)) is discretized using Eq. (2.24):

$$\frac{\psi_i^{n^*+1} - \psi_i^{n^*}}{\Delta\tau} = \frac{1}{V_i} \sum_{k\in\mathcal{N}_i} \left(\frac{1}{4\cosh^2\left(\frac{\phi_{\mathrm{map},ik}^{n^*}}{2\varepsilon_{ik}}\right)} \left(\boldsymbol{\nabla}\phi_{\mathrm{map}}|_{ik}^{n^*} \cdot \mathbf{n}_{ik} - \mathbf{n}_{ik} \cdot \mathbf{n}_{ik} \right) \right) \mathbf{n}_{ik} \cdot \mathbf{dA}_{ik} \tag{2.31}$$

where $\phi_{\mathrm{map},ik} = (\phi_{\mathrm{map},i} + \phi_{\mathrm{map},k})/2$ and $\varepsilon_{ik} = (\varepsilon_i + \varepsilon_k)/2$ are linearly interpolated interface distance and profile thickness, respectively. $\boldsymbol{\nabla}\phi_{\mathrm{map}}|_{ik}$ is computed using a compact term in the direction of the node pair as in Eq. (2.27):

$$\begin{aligned} \boldsymbol{\nabla}\phi_{\mathrm{map}}|_{ik} = {} & \frac{\phi_{\mathrm{map},k} - \phi_{\mathrm{map},i}}{\Delta x_{ik}} \mathbf{e}_{ik} \\ & + \frac{\boldsymbol{\nabla}\phi_{\mathrm{map}}|_i^{2\mathrm{nd}} + \boldsymbol{\nabla}\phi_{\mathrm{map}}|_k^{2\mathrm{nd}}}{2} - \left(\frac{\boldsymbol{\nabla}\phi_{\mathrm{map}}|_i^{2\mathrm{nd}} + \boldsymbol{\nabla}\phi_{\mathrm{map}}|_k^{2\mathrm{nd}}}{2} \cdot \mathbf{e}_{ik} \right) \mathbf{e}_{ik} \end{aligned} \tag{2.32}$$

where $\boldsymbol{\nabla}\phi_{\mathrm{map}}|_{i\,\mathrm{or}\,k}^{2\mathrm{nd}}$ are second-order nodal gradients computed using Eq. (2.22).

The interface normals $\mathbf{n}_{ik}$ are computed with Eq. (2.25). It has been chosen to use non-unit normals, to avoid spurious gradient orientations when several interfaces are close to each other, and thus improve the front-merging properties of the implemented reinitialization.

A particular numerical treatment is needed for the nodes of the narrow band where ψ is close to 0 or 1. Indeed, at these nodes, ϕ_{map} is difficult to estimate and two improvements in the reinitialization algorithm are introduced: (i) in order to ensure a correct computation of $\boldsymbol{\nabla}\phi_{\mathrm{map}}|_{ik}$, which requires several nodal values of ϕ_{map}, an under/overshoot tag is attributed to all pairs with at least one node concerned by ψ falling outside $]0,1[$; (ii) the resharpening term is removed so that the reinitialization becomes purely diffusive. Using the original isotropic diffusion term of Olsson and Kreiss [29] in this case, the reinitialization then takes the form:

$$\frac{\psi_i^{n^*+1} - \psi_i^{n^*}}{\Delta\tau} = \frac{1}{V_i} \sum_{k\in\mathcal{N}_i} \varepsilon_{ik} \boldsymbol{\nabla}\psi|_{ik}^{n^*} \cdot \mathbf{dA}_{ik} \tag{2.33}$$

with $\boldsymbol{\nabla}\psi|_{ik}^{n^*}$ discretized as in Eq. (2.32). Once the overshoots or undershoots of ψ disappear, the reinitialization turns back to its main form given by Eq. (2.31).

Usually, 3 steps of reinitialization are performed and the pseudo-timestep of each reinitialization step is simply determined by a Fourier condition on the diffusive part of the reinitialization. This stability condition leads to:

$$\Delta\tau = \min_{\text{pairs } ik}\left(\frac{\text{Fo}.\Delta x_{ik}^2}{\varepsilon_{ik}}\right) \tag{2.34}$$

where Δx_{ik} stands for the length of the pair ik, and Fo is the Fourier number set to 0.5. Hence, with $\varepsilon_{ik} \sim \Delta x_{ik}$, $\Delta\tau \sim \Delta x_{ik}$.

2.2.7 Transport of the ACLS function - Time integration

The ACLS function ψ is advected using Eq. (2.4). In YALES2, a 4th-order discretization is performed for convective terms, leading to the following semi-discrete equation for a node i:

$$\frac{\partial\psi_i}{\partial t} + \boldsymbol{\nabla}\cdot(\psi\mathbf{u})\Big|_i = 0 \tag{2.35}$$

Then by applying Eq. (2.24):

$$\frac{\partial\psi_i}{\partial t} + \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\psi_{ik}\mathbf{u}_{ik}\cdot\mathbf{dA}_{ik} = 0 \tag{2.36}$$

Using a 4th-order spatial discretization as in Section 2.2.3, Eq. (2.36) yields:

$$\frac{\partial\psi_i}{\partial t} + \frac{1}{V_i}\sum_{k\in\mathcal{N}_i}\left(\frac{\psi_i+\psi_k}{2} + \frac{\boldsymbol{\nabla}\psi|_i^{\text{2nd}} - \boldsymbol{\nabla}\psi|_k^{\text{2nd}}}{6}\cdot\boldsymbol{\Delta x}_{ik}\right) \left(\frac{\mathbf{u}_i+\mathbf{u}_k}{2} + \frac{\boldsymbol{\nabla}\mathbf{u}|_i^{\text{2nd}} - \boldsymbol{\nabla}\mathbf{u}|_k^{\text{2nd}}}{6}\cdot\boldsymbol{\Delta x}_{ik}\right)\cdot\mathbf{dA}_{ik} = 0 \tag{2.37}$$

Time integration is performed using the TFV4A scheme of [84], which is a 4th-order, Runge-Kutta-like, time-marching scheme: see Section 1.2.5 for more details. The timestep Δt is computed from the classical explicit CFL stability criterion:

$$\Delta t = \min_{\text{pairs } ik}\left(\frac{\text{CFL}|\mathbf{dA}_{ik}|\Delta x_{ik}}{|\mathbf{u}_{ik}\cdot\mathbf{dA}_{ik}|}\right) \tag{2.38}$$

where Δx_{ik} is the length of the pair ik. The full ACLS procedure is depicted in Fig. 2.6.

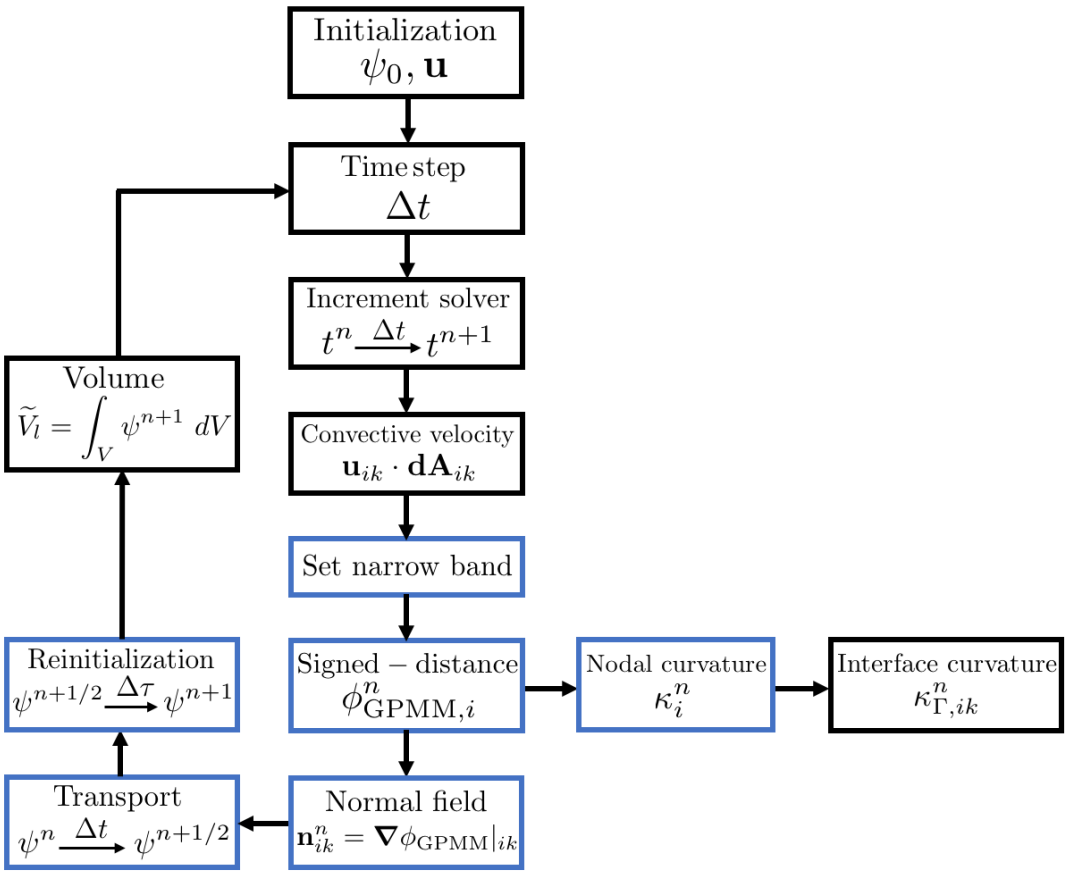


Figure 2.6: Full ACLS procedure: the blue boxes represent computations in the narrow band.

2.3 Incompressible flow solver

To solve the incompressible Navier-Stokes equations, a projection method is employed [57]: it is thus needed to solve a Poisson equation accounting for the interfacial pressure jump.

2.3.1 Velocity prediction with momentum correction

In order to avoid velocity mixing between liquid and gas, and thus limit spurious kinetic energy transfers at the interface, Desjardins and Moureau introduced a momentum correction at the interface in [166]. Indeed, two sources of non-conservation of momentum are identified in the projection procedure: (i) the velocity prediction which needs to be tightly coupled to interface transport through density; (ii) the correction step as it is written in non-conservative form. The dominant error regards the transport step (i): the strategy detailed in this section is thus to limit the momentum-conservation errors at the prediction step. In this section, we only consider the convection term in Eq. (2.10) for convenience. The prediction step writes:

$$\frac{\mathbf{u}^* - \mathbf{u}^n}{\Delta t} = -\boldsymbol{\nabla} \cdot \left(\mathbf{u}^{n+1/2} \otimes \mathbf{u}^{n+1/2} \right) \tag{2.39}$$

which can be rewritten in terms of momentum:

$$\frac{\rho^{**}\mathbf{u}^* - \rho^n \mathbf{u}^n}{\Delta t} = -\boldsymbol{\nabla} \cdot \left(\rho^{n+1/2} \mathbf{u}^{n+1/2} \otimes \mathbf{u}^{n+1/2} \right) \tag{2.40}$$

where ρ^{**} is a predicted density computed from the following step:

$$\frac{\rho^{**} - \rho^n}{\Delta t} = -\boldsymbol{\nabla} \cdot \left(\rho^{n+1/2} \mathbf{u}^{n+1/2} \right) \tag{2.41}$$

Eq. (2.41) can be discretized in space as:

$$\frac{\rho_i^{**} - \rho_i^n}{\Delta t} = -\frac{1}{V_i} \sum_{k \in \mathcal{N}_i} \rho_{ik}^{n+1/2} \mathbf{u}_{ik} \cdot \mathbf{dA}_{ik} \tag{2.42}$$

with $\rho_{ik}^{n+1/2}$ computed using a simple upwinded scheme to ensure the boundedness of the density flux at the interface:

$$\rho_{ik}^{n+1/2} = \begin{cases} (1 - \mathcal{U}_{ik}) \dfrac{\rho_i^n + \rho_k^n}{2} + \mathcal{U}_{ik} \rho_i^n & \text{if } \mathbf{u}_{ik} \cdot \mathbf{dA}_{ik} \geq 0 \\ (1 - \mathcal{U}_{ik}) \dfrac{\rho_i^n + \rho_k^n}{2} + \mathcal{U}_{ik} \rho_k^n & \text{if } \mathbf{u}_{ik} \cdot \mathbf{dA}_{ik} < 0 \end{cases} \tag{2.43}$$

where the pair-based limiter $\mathcal{U}$ is computed as:

$$\mathcal{U}_{ik} = \max \left(\frac{|b|_{\text{upw}} - \min\left(|b_i|, |b_k|\right) + 1}{|b|_{\text{upw}}}, 0 \right) \tag{2.44}$$

with $|b|_{\text{upw}} = 5$, meaning that the first five narrow-band levels from the interface are concerned by the upwinding, which decreases when increasing the band number. A similar numerical strategy is applied for the conservative flux of Eq. (2.40):

$$\frac{\rho_i^{**}\mathbf{u}_i^* - \rho_i^n \mathbf{u}_i^n}{\Delta t} = -\frac{1}{V_i} \sum_{k \in \mathcal{N}_i} \rho \mathbf{u} \Big|_{ik}^{n+1/2} \mathbf{u}_{ik} \cdot \mathbf{dA}_{ik} \tag{2.45}$$

with $\rho\mathbf{u}\Big|_{ik}^{n+1/2}$ calculated to 4th-order as:

$$\rho\mathbf{u}\Big|_{ik}^{n+1/2} = \begin{cases} (1-\mathcal{U}_{ik})\dfrac{\rho_i^n\mathbf{u}_i^n+\rho_k^n\mathbf{u}_k^n}{2}+\mathcal{U}_{ik}\rho_i^n\mathbf{u}_i^n+\rho_{ik}^{n+1/2}\dfrac{\boldsymbol{\nabla}\mathbf{u}|_i^{2\text{nd}}-\boldsymbol{\nabla}\mathbf{u}|_k^{2\text{nd}}}{6}\cdot\boldsymbol{\Delta x}_{ik} \\ \qquad\qquad \text{if } \mathbf{u}_{ik}\cdot\mathbf{dA}_{ik}\geq 0 \\ (1-\mathcal{U}_{ik})\dfrac{\rho_i^n\mathbf{u}_i^n+\rho_k^n\mathbf{u}_k^n}{2}+\mathcal{U}_{ik}\rho_k^n\mathbf{u}_k^n+\rho_{ik}^{n+1/2}\dfrac{\boldsymbol{\nabla}\mathbf{u}|_i^{2\text{nd}}-\boldsymbol{\nabla}\mathbf{u}|_k^{2\text{nd}}}{6}\cdot\boldsymbol{\Delta x}_{ik} \\ \qquad\qquad \text{if } \mathbf{u}_{ik}\cdot\mathbf{dA}_{ik}<0 \end{cases} \tag{2.46}$$

The proposed method is not stated as mass-momentum consistent, but allows momentum conservation over Δt at the prediction step (no velocity mixing between gas and liquid). A momentum-conservation error appears when ρ^{**} is discarded and replaced by ρ^{n+1}: this error however remains smaller than errors at the prediction step. Mass-momentum consistency, which is a required feature to robustly simulate complex high-density ratio two-phase flows as mentioned in [169], [170] or [171], is left for future work.

2.3.2 Ghost-Fluid Method

The Ghost-Fluid Method [31, 37] is a robust and accurate way to explicitly deal with jump conditions. Let us consider the jump at a liquid-gas interface in one dimension of a physical quantity ζ, written $[\zeta]_\Gamma = [\zeta](x_\Gamma) = \zeta_l(x_\Gamma) - \zeta_g(x_\Gamma) = \zeta_{l,\Gamma} - \zeta_{g,\Gamma}$. The liquid and gas quantities in each phase can be extended in the other phase by continuity:

$$\zeta_l(x) = \zeta_{l,\Gamma} + (x-x_\Gamma)\frac{\partial\zeta}{\partial x}\Big|_{l,\Gamma} + \mathcal{O}(x-x_\Gamma)^2 \text{ and } \zeta_g(x) = \zeta_{g,\Gamma} + (x-x_\Gamma)\frac{\partial\zeta}{\partial x}\Big|_{g,\Gamma} + \mathcal{O}(x-x_\Gamma)^2 \tag{2.47}$$

which leads to the expression of the jump everywhere in the fluid domain:

$$[\zeta](x) = [\zeta]_\Gamma + (x-x_\Gamma)\left[\frac{\partial\zeta}{\partial x}\right]_\Gamma + \mathcal{O}(x-x_\Gamma)^2 \text{ with } \left[\frac{\partial\zeta}{\partial x}\right]_\Gamma = \frac{\partial\zeta}{\partial x}\Big|_{l,\Gamma} - \frac{\partial\zeta}{\partial x}\Big|_{g,\Gamma} \tag{2.48}$$

The extension to 3D is straightforward:

$$[\zeta](\mathbf{x}) = [\zeta]_\Gamma + (\mathbf{x}-\mathbf{x}_\Gamma)\cdot[\boldsymbol{\nabla}\zeta]_\Gamma + \mathcal{O}(\mathbf{x}-\mathbf{x}_\Gamma)^2 \tag{2.49}$$

Fig. 2.7 illustrates the principle of the Ghost-Fluid Method. In the following, this methodology is used to discretize the pressure gradient in the numerical framework of this study: pair-based finite-volume schemes.

2.3.2.1 Discretization of the pressure gradient

A first-order finite-difference approximation of the pressure term gives:

$$\frac{1}{\rho}\frac{\partial p}{\partial x}\Big|_{l,i+\frac{1}{2}} = \frac{1}{\rho_l}\frac{p_{l,i+1}-p_{l,i}}{\Delta x} + \mathcal{O}(\Delta x) \tag{2.50}$$

Using the pressure jump at x_i, the liquid pressure is $p_{l,i} = [p]_i + p_{g,i}$ and can be replaced as follows:

$$\frac{1}{\rho}\frac{\partial p}{\partial x}\Big|_{l,i+\frac{1}{2}} = \frac{1}{\rho_l}\frac{p_{l,i+1}-[p]_i-p_{g,i}}{\Delta x} + \mathcal{O}(\Delta x) \tag{2.51}$$

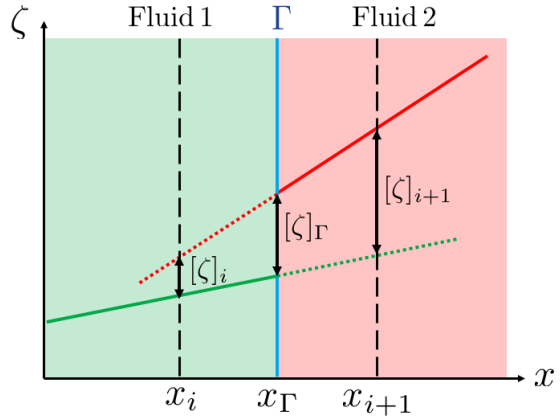


Figure 2.7: Ghost-Fluid Method: extension of the jump of a physical quantity ζ.

The expression for the pressure jump $[p]_i$ is given by Eq. (2.48):

$$[p]_i = [p]_\Gamma + (x_i - x_\Gamma)\left[\frac{\partial p}{\partial x}\right]_\Gamma + \mathcal{O}(x_i - x_\Gamma)^2 \tag{2.52}$$

$[p]_\Gamma$ is a known quantity (Eq. (2.11)) but $[\partial p/\partial x]_\Gamma$ is unknown a priori. With the assumption that the velocity is continuous across the interface, the pressure term in the momentum equation Eq. (2.10) should also be continuous:

$$\left[\frac{1}{\rho}\frac{\partial p}{\partial x}\right]_\Gamma = 0 \Longleftrightarrow \frac{1}{\rho_l}\frac{\partial p}{\partial x}\Big|_{l,\Gamma} = \frac{1}{\rho_g}\frac{\partial p}{\partial x}\Big|_{g,\Gamma} \tag{2.53}$$

After many developments, which can be found in [31], the final form of the discretized pressure gradient for finite differences is obtained, for $\Gamma \in [x_i; x_{i+1}]$:

$$\frac{1}{\rho}\frac{\partial p}{\partial x}\Big|_{l,i+\frac{1}{2}} = \frac{1}{\rho}\frac{\partial p}{\partial x}\Big|_{g,i+\frac{1}{2}} = \frac{1}{\rho^*}\frac{p_{l,i+1} - p_{g,i}}{\Delta x} - \frac{1}{\rho^*}\frac{[p]_\Gamma}{\Delta x} + \mathcal{O}(\Delta x) \tag{2.54}$$

with $\rho^* = \theta\rho_g + (1-\theta)\rho_l$ a modified density, computed using the interpolation weight defined in Eq. (2.15).

2.3.2.2 Implementation in the Poisson solver

YALES2 is a pair-based finite-volume solver, as a result implementation of finite-difference schemes can be performed on the pairs. The expression of the discretized pressure term of the momentum equation written in Eq. (2.54) is integrated in the Poisson solver of YALES2 to account for interfacial jump conditions.

To solve the incompressible Navier-Stokes equations, a projection method is employed [57]. The Poisson equation, used to enforce the solenoidal nature of the velocity field $\mathbf{u}^{n+1}$, is:

$$\boldsymbol{\nabla}\cdot\left(\frac{1}{\rho}\boldsymbol{\nabla}p\right) = \frac{1}{\Delta t}\boldsymbol{\nabla}\cdot\mathbf{u}^* \tag{2.55}$$

with $\mathbf{u}^*$ the predicted velocity, estimated from $\mathbf{u}^n$ (see previous Section 2.3.1). Eq. (2.55) can be discretized as presented in Section 2.2.3:

$$\sum_{k\in\mathcal{N}_i}\frac{1}{\rho}\boldsymbol{\nabla}p\Big|_{ik}\cdot\mathbf{dA}_{ik} = \frac{1}{\Delta t}\sum_{k\in\mathcal{N}_i}\mathbf{u}^*_{ik}\cdot\mathbf{dA}_{ik} \tag{2.56}$$

in which the pressure gradient is computed at pairs. A linear system of the form $Ap = b$ that includes the jump conditions needs to be solved, and is written as:

$$\underbrace{\sum_{k\in\mathcal{N}_i} \frac{1}{\rho^*}\frac{p_k - p_i}{\Delta x_{ik}}\mathbf{e}_{ik}\cdot\mathbf{dA}_{ik}}_{Ap} = \underbrace{\frac{1}{\Delta t}\sum_{k\in\mathcal{N}_i}\mathbf{u}^*_{ik}\cdot\mathbf{dA}_{ik} + \sum_{k\in\mathcal{N}_i}\frac{1}{\rho^*}\frac{[p]_\Gamma}{\Delta x_{ik}}\mathbf{e}_{ik}\cdot\mathbf{dA}_{ik}}_{b} \tag{2.57}$$

The second part of the RHS b, which contains the pressure jump, is added for all pairs ik crossing the interface. The linear system is solved using the Deflated Preconditioned Conjugate Gradient of [59].

The correction step is written as:

$$\frac{\mathbf{u}^{n+1} - \mathbf{u}^*}{\Delta t} = -\frac{1}{\rho}\boldsymbol{\nabla} p\Big|_i \tag{2.58}$$

in which the pressure gradient is calculated at nodes. For a pair of nodes which does not cross the interface (i.e. within one incompressible phase, liquid or gas), the pressure gradient is simply discretized as in Section 2.2.3, with a constant density ρ corresponding to the phase. For a crossing pair ik, the jump conditions need to be accounted for:

$$\frac{1}{\rho}\boldsymbol{\nabla} p\Big|_i^{\text{2nd}} = \frac{1}{\rho_i V_i}\sum_{k\in\mathcal{N}_i}\frac{p_i + p_k + [p]_k}{2}\mathbf{dA}_{ik} \tag{2.59}$$

where the pressure jump is expressed from the developments above using the Ghost-Fluid Method:

$$[p]_k = \frac{\rho_i}{\rho^*}[p]_\Gamma + \left(1 - \frac{\rho_i}{\rho^*}\right)(p_i - p_k) \tag{2.60}$$

2.3.3 Time integration - Capillary timestep restriction

As in Section 2.2.7, time integration is performed using the TFV4A scheme of [84], which is a 4th-order, Runge-Kutta-like, time-marching scheme: see Section 1.2.5 for more details. The timestep Δt is computed from the explicit capillary constraint:

$$\Delta t = \min_{\text{crossing pairs } ik}\sqrt{\frac{\text{STN}^2\Delta x_{ik}^2\min(\rho_g,\rho_l)}{\sigma|\kappa_{\Gamma,ik}|}} \tag{2.61}$$

where STN is the surface tension number, usually set to 0.5.

2.4 Isotropic dynamic mesh adaptation strategy

As introduced in Section 1.2.3, to resolve the small physical scales at the liquid-gas interface on tetrahedral grids at a moderate cost, an isotropic Adaptive Mesh Refinement (AMR) technique is used [78], employing the MMG sequential remeshing library [77]. The parallelism is handled by the YALES2 flow solver (interpolation, data transfer), so that the dynamic mesh adaptation strategy is tailored for massively-distributed grids.

2.4.1 Target metric generation

Special attention is needed when coupling AMR with the ACLS interface-capturing method to avoid liquid mass losses, so that the interface Γ never encounters cell-size gradients. Thus, the mesh is refined around the interface based on the signed distance $\mathcal{D} = \phi$ to it, so that Γ always stays in a protected region of constant cell size (or metric) $\Delta x_{\min}$, defined by the user. To this aim, a target metric $\mathcal{M}$ is generated at nodes as a function of the reconstructed signed-distance function $\mathcal{D} = \phi_{\text{GPMM}}$ (see Fig. 2.8 for spatial evolution of $\mathcal{M}$) [6]:

$$\mathcal{M}(\mathcal{D}) = \min\Big(\Delta x_{\text{init}}, \max\left(\Delta x_{\min}, (|\mathcal{D}| - N_p \Delta x_{\min}) \times |\boldsymbol{\nabla}\mathcal{M}|_0\right)\Big) \tag{2.62}$$

where Δx_{init} is the initial coarse cell size, $|\boldsymbol{\nabla}\mathcal{M}|_0$ is the prescribed cell-size gradient, defined by the user, typically set to 1.3, to ensure a smooth and controlled linear transition of the metric from the finest cells of size $\Delta x_{\min}$ to the coarsest ones of size Δx_{init}. $N_p \Delta x_{\min}$ is the half-width of the refined region, with N_p a user-defined parameter, usually set between 6 and 12. Indeed, the choice of N_p drives the size of the adaptive grid and the remeshing frequency. Increasing the value of N_p leads in turn to an increase in the number of mesh cells (wider refined zone), but one needs to remesh less often in order to keep the interface in the finest region (less AMR calls). Conversely, reducing the value of N_p diminishes the number of grid cells (thiner refined zone), but remeshing is needed more frequently (more AMR calls). In our massively-parallel two-phase flow simulations, $6 < N_p < 12$ seems a good compromise between mesh size and adaptation frequency. It is important to stress that this metric evolution ensures that the remeshing process

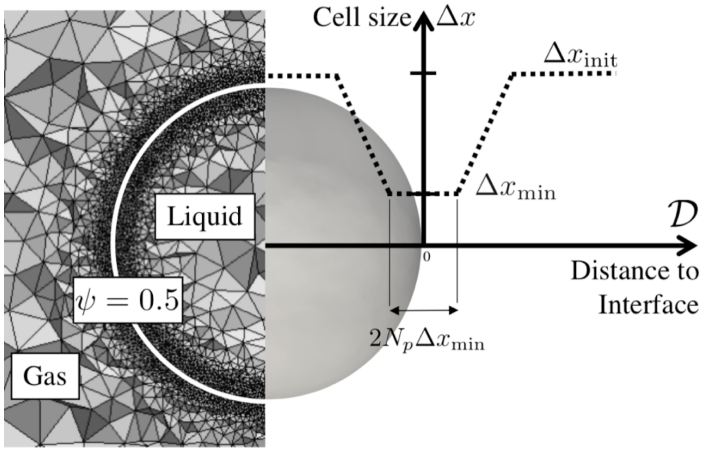


Figure 2.8: Adaptive mesh refinement strategy: refinement based on the distance $\mathcal{D}$ to the interface.

occurs at the vicinity of the interface but never at the interface itself. This strategy avoids any interpolation error at the interface location. It should also be noted that as the reconstructed GPMM signed-distance ϕ_{GPMM} is needed far from the interface to build the target metric in Eq. (2.62), the narrow band is set three times wider when adaptation is triggered at a given solver iteration. For iterations without AMR, the classical value is used (see Section 2.2.1).

2.4.2 Automatic triggering

In practice, the adaptation process is triggered automatically when the interface is a few cells away from the edge of the fine region. This is done by the mean of a user-defined protected distance

ϕ_p, which determines the width of a protected region within the fine zone. This width is typically half the one of the refined region: $|\phi_p| = N'_p \Delta x_{\min} = (N_p/2)\Delta x_{\min}$. It thus means that the interface should always have at least N'_p fine cells of size $\Delta x_{\min}$ ahead of it. The AMR process is triggered each time at least one node i belonging to the protected region (i.e. $|\phi_{\mathrm{GPMM},i}| < |\phi_p|$) falls outside the refined region. The automatic distance-based triggering strategy for a perfectly straight interface is illustrated in Fig. 2.9. This automatic triggering leads to considerable computational savings in remeshing compared to the use of a constant user-defined frequency, as it adapts dynamically to the instantaneous displacement of the interface.

2.4.3 Parallel iterative remeshing

The parallelism of the dynamic mesh adaptation procedure relies on the double-domain decomposition of the YALES2 flow solver, presented in Section 1.2.6: grid cells are gathered in cell groups (typically 500 cells per group: $N_{\mathrm{cells/group}} = 500$), themselves gathered within a processor [38]. The treatment of the interface between processors is delicate when performing dynamic mesh adaptation: one has to deal with bad quality cells at these interfaces. The moving interface method, generalized to a two-level domain decomposition [78, 80], is used to address this issue: the regions near the processor boundaries are firstly left untouched, which yields a local adaptation step without any communications between ranks, calling the MMG sequential library [77]. This induces bad quality cells, identified by their skewness, at processor interfaces (see Fig. 2.10).
Then, interfacial element groups are moved so that the previous processor interface becomes immersed within the new partition, which will be then adapted in turn. This procedure is done iteratively until no bad quality cells remain in the domain. The scheduling of the movement of cell groups is driven by a constrained load balancing algorithm using the METIS library [85]. This operation ensures that the number of tetrahedral cells remains properly balanced between ranks. The two-level parallel dynamic grid adaptation technique is illustrated in Fig. 2.11.
The full parallel dynamic mesh adaptation procedure can then be detailed as [6, 80]:

(i) Advancement of the liquid-gas interface using the unstructured ACLS procedure;

(ii) Generation of the target metric $\mathcal{M}$ given by Eq. (2.62);

(iii) For each processor, cell groups are merged into a single one;

(iv) Each MPI process calls the MMG sequential library [77] to adapt each merged group provided the target metric $\mathcal{M}$. The nodes at merged group boundaries (processor interfaces or walls) are frozen to preserve the connectivity of processor interfaces and minimize parallel communications. The drawback is the generation of bad quality cells at processor interfaces, measured by their skewness $\mathcal{S}$;

(v) For each processor, adapted merged groups are split into cell groups using the METIS coloring library [85], so that each group contains $N_{\mathrm{cells/group}}$, and data are interpolated on the new mesh;

(vi) The two-level moving interface method of [78] is employed to deal with bad quality cells: the location of processor interfaces is modified iteratively and sequential load balancing is performed. The element groups to be transfered are selected based on the skewness of the cells at the processor interface. Thus, the skewness $\mathcal{S}$ of each cell $\mathcal{C}$ with at least one node connected to the interface γ between the local MPI process $\mathcal{P}_{\mathrm{in}}(\gamma)$ and its neighbor $\mathcal{P}_{\mathrm{out}}(\gamma)$

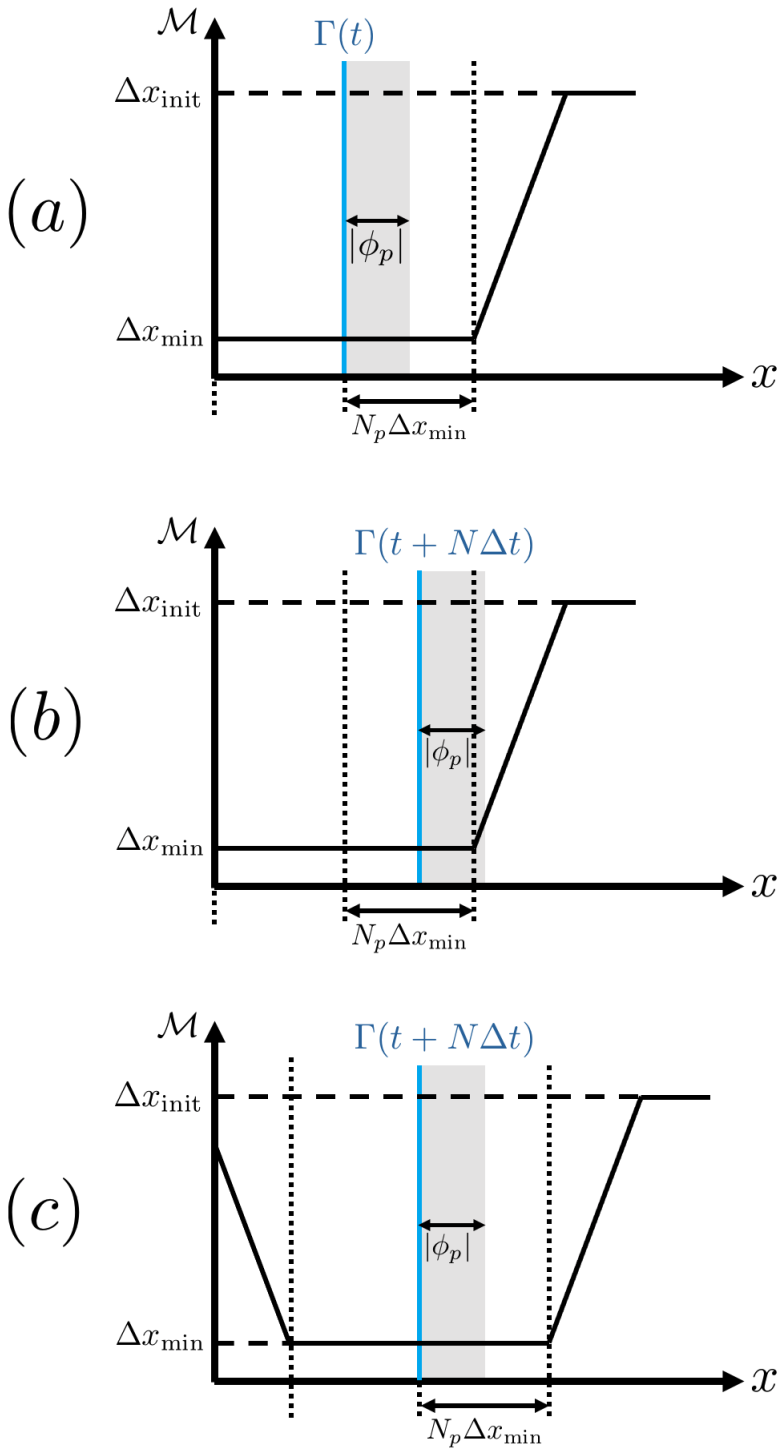


Figure 2.9: 1D illustration of the AMR triggering using a user-defined protected distance $|\phi_p| = (N_p/2)\Delta x_{\text{min}}$: (a) Liquid-gas interface Γ right after an adaptation process (instant t); (b) Interface N fluid iterations later (instant $t+N\Delta t$), triggering of AMR; (c) Remeshing around Γ at $t+N\Delta t$.

is computed. This skewness is compared to the maximum allowed skewness $\mathcal{S}_{\text{max}}$, defined by the user (typically 0.99). If bad quality cells are present at processor interface γ, i.e.:

$$\max_{\mathcal{C}} \left(\mathcal{S}(\mathcal{C}, \gamma)\right) > \mathcal{S}_{\text{max}} \tag{2.63}$$

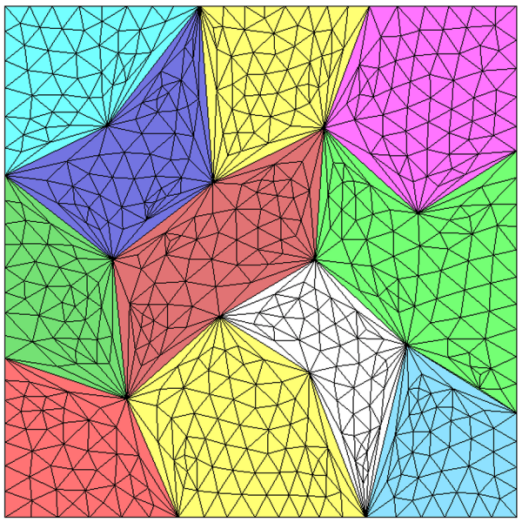


Figure 2.10: Bad-quality grid cells at interfaces between processors. Courtesy of A. Froehly - ParMMG [77].

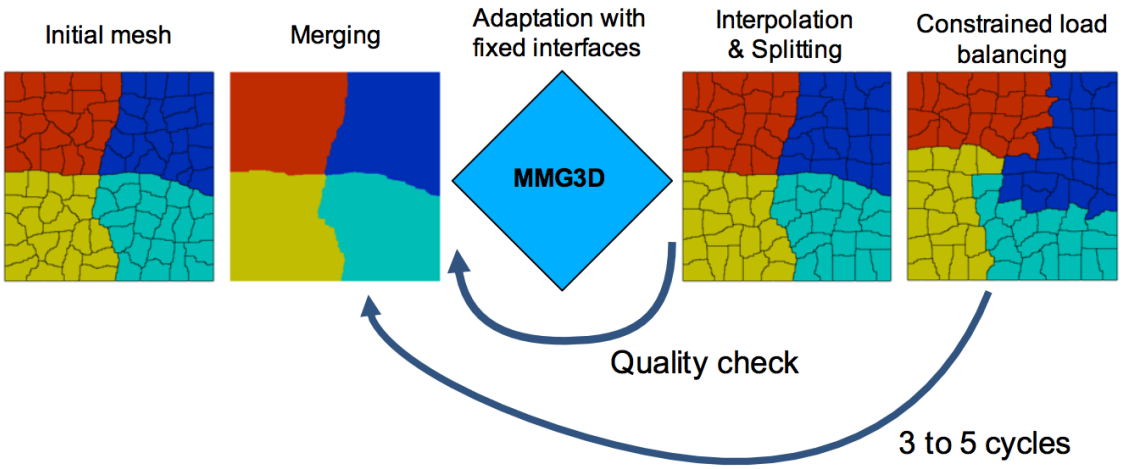


Figure 2.11: Schematic of the two-level moving interface parallel adaptation strategy for 4 processors. The thin black lines represent the interfaces between cell groups and the different colors correspond to the processor ranks. Reprinted from [80].

the associated cell group at the boundary of $\mathcal{P}_{\text{in}}(\gamma)$ is transfered to $\mathcal{P}_{\text{out}}(\gamma)$, and the interfacial cells are thus immersed into the $\mathcal{P}_{\text{out}}(\gamma)$ partition. This approach drastically decreases the number of transfered element groups and therefore increases the efficiency of the adaptation step. Once the element groups have been transfered, the element and element group connectivities are rebuilt;

(vii) Steps (iii) to (vi) are repeated to iteratively adapt the mesh, until the maximum skewness of the full grid is below $\mathcal{S}_{\max}$. In addition to the skewness, a local metric error $\varepsilon_{\mathcal{M}}$ is also computed, relatively to the target metric. The maximum relative metric error in the domain should be lower than a user-defined threshold $\varepsilon_{\mathcal{M},\max}$, typically set between 30% and 120% of the target metric. If both the maximum skewness and relative metric error criteria are satisfied, the grid is considered as adapted and the fluid calculation continues with the new grid, until the next automatic AMR triggering (Section 2.4.2).

CHAPTER 3

Validation on unstructured grids

Disclaimer: Part of this chapter has been published in *Journal of Computational Physics* [216].

Contents

3.1 Accuracy of signed-distance and curvature computation 77
3.2 Interface transport tests 82
3.2.1 Static sphere on static tetrahedral grids 82
3.2.2 Rotation of a notched disk on static triangular grids 82
3.2.3 Circle deformation in a vortex on static triangular grids 84
3.2.4 Sphere deformation in a vortex on adaptive tetrahedral grids 86
3.3 Capillary-driven flows 89
3.3.1 Damping of a capillary wave on static triangular grids 89
3.3.2 Spurious currents on static tetrahedral grids 92
3.3.3 Rayleigh-Plateau instability on static tetrahedral grids 93
3.3.4 3D droplet collision using adaptive mesh refinement 96

The present chapter aims to validate the various parts of the YALES2 two-phase flow solver presented in the previous chapter, using static and adaptive unstructured meshes. The accuracy and convergence of the GPMM signed-distance reconstruction and of the interface curvature computation, a central matter in interfacial flows, are first assessed both in 2D and 3D for circular and spherical interfaces, respectively. Then, classical interface-transport tests are performed with quantified error levels and spatial convergence. Eventually, the ability of the solver to properly simulate surface-tension-driven flows and to implicitly predict topology changes (breakup and merging; which is not guaranteed when discretizing the latest reinitialization equation), is checked through canonical two-phase flow examples.

3.1 Accuracy of signed-distance and curvature computation

A 2D/3D static droplet of radius R_d is initialized on homogeneous triangular/tetrahedral grids to assess the accuracy of the GPMM procedure, both in 2D and 3D, for the signed-distance reconstruction on irregular meshes. The test case, which will also be used to assess the accuracy of the interface curvature computation, is illustrated in Fig. 3.1. Errors are calculated as follows:

$$L_2(\phi) = \sqrt{\frac{1}{N_{i\in|b|}} \sum_{i=1}^{N_{i\in|b|}} \left(\phi_{\mathrm{th},i} - \phi_{\mathrm{GPMM},i}\right)^2} \quad \text{and} \quad L_\infty(\phi) = \max_{i\in|b|} \left|\phi_{\mathrm{th},i} - \phi_{\mathrm{GPMM},i}\right| \tag{3.1}$$

where $|b|$ is the absolute narrow-band level, $\phi_{\mathrm{th},i}$ is the theoretical distance at node i to the circular/spherical interface, and $N_{i\in|b|}$ is the total number of nodes in the corresponding absolute narrow-band level. The calculations of L_2 and L_∞ are performed for nodes belonging to edges

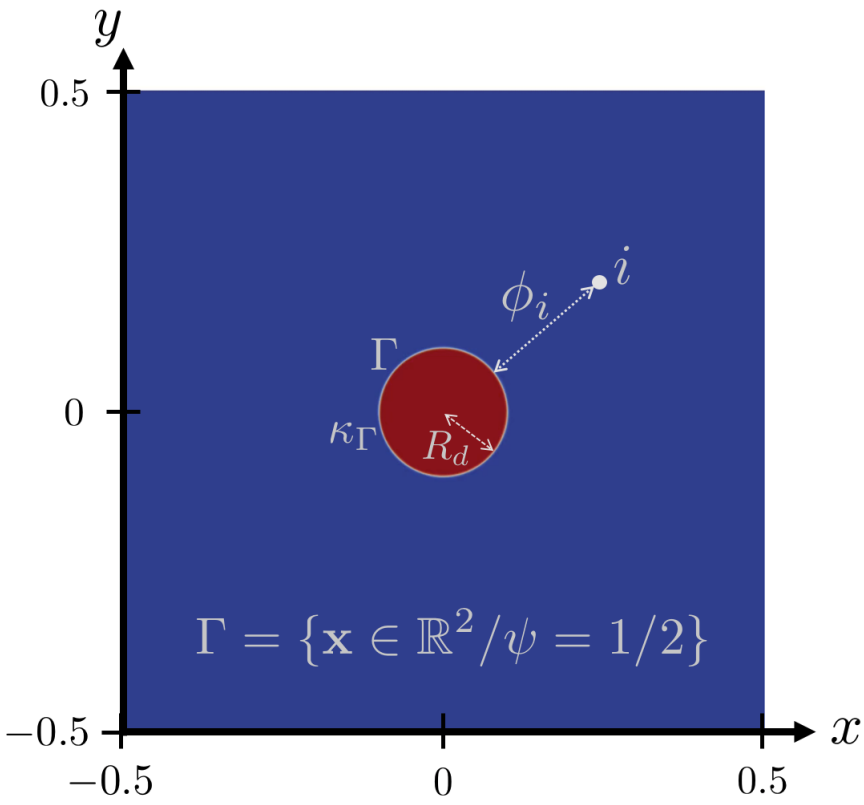


Figure 3.1: Static droplet of radius R_d in a squared domain. The signed distance from a node i to the interface Γ is ϕ_i. κ_Γ is the curvature at the interface.

crossed by the interface ($|b| = 1$) and in the sixth band level ($|b| = 6$, far from the interface). As homogeneous triangles/tetrahedra are used to tessellate the domain, the mesh cell size is computed as $\Delta x = \sqrt{V_i}$ in 2D, and as $\Delta x = \sqrt[3]{V_i}$ in 3D, where V_i is the control volume based on the dual mesh around a node i. Results are shown in Fig. 3.2 and Fig. 3.3 for the 2D and 3D cases, respectively.

It can be seen in Fig. 3.2 (a) that in the first band level ($|b| = 1$), i.e. close to the interface, the GPMM procedure converges with second-order accuracy (L_2 and L_∞) with low error values, which is satisfactory for triangular grids. Fig. 3.2 (b) shows the results for the sixth band level ($|b| = 6$), i.e. far from the interface. Again, as the resolution of the droplet increases, both L_2 and L_∞ errors exhibit second-order convergence. The errors and corresponding order of convergence for 3D tetrahedral meshes are displayed in Fig. 3.3 and Table 3.1: second-order convergence is shown for the L_2 norm, while the L_∞ error displays a convergence closer to first order. As several markers are propagated in the narrow band ($N_M = 3$ in 2D and $N_M = 10$ in 3D, see Section 2.2.2), a sufficient amount of data is available at a node far from the interface to compute the projected distance to the closest surface element. The advantage of the GPMM algorithm, which is an exact geometric projection, is that the numerical errors are only induced by segmentation (2D) or triangulation (3D): a perfect line (2D) or plane (3D) on an unstructured mesh leads to a distance computed within machine accuracy. A way to lower the errors even more could be to improve the sub-grid representation of the interface, such as attaching Lagrangian particles to the interface, similarly to Front-Tracking techniques, and to compute the distance at nodes in the narrow band to the closest particles [172].

On the same test case, the accuracy of the interface curvature κ_Γ (Eq. (2.29)) is evaluated in

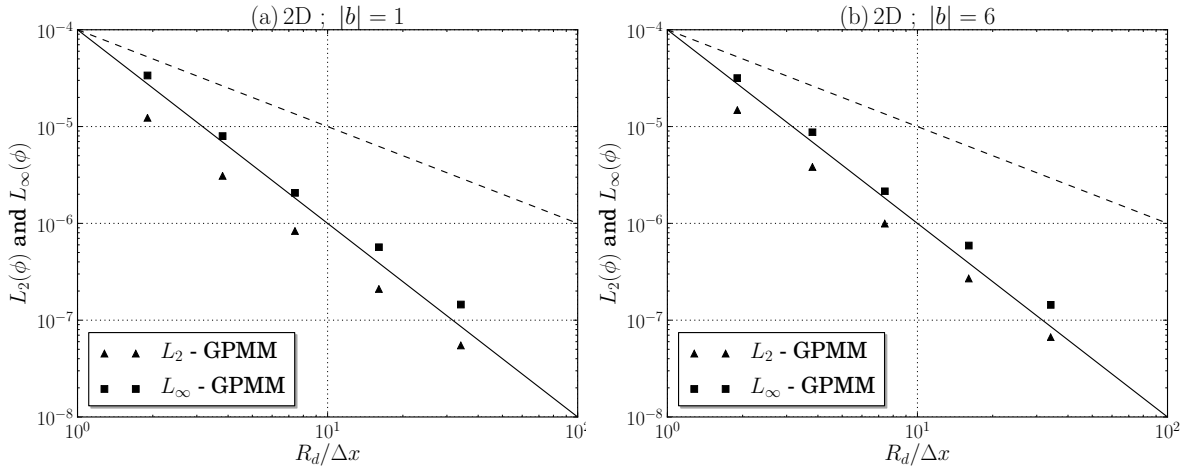


Figure 3.2: L_2 and L_∞ of the reconstructed signed-distance function ϕ with the 2D GPMM algorithm (Geometric-Projection Marker Method), in first (a) and sixth (b) narrow-band levels $|b|$. The dashed and solid lines are first and second-order convergence, respectively.

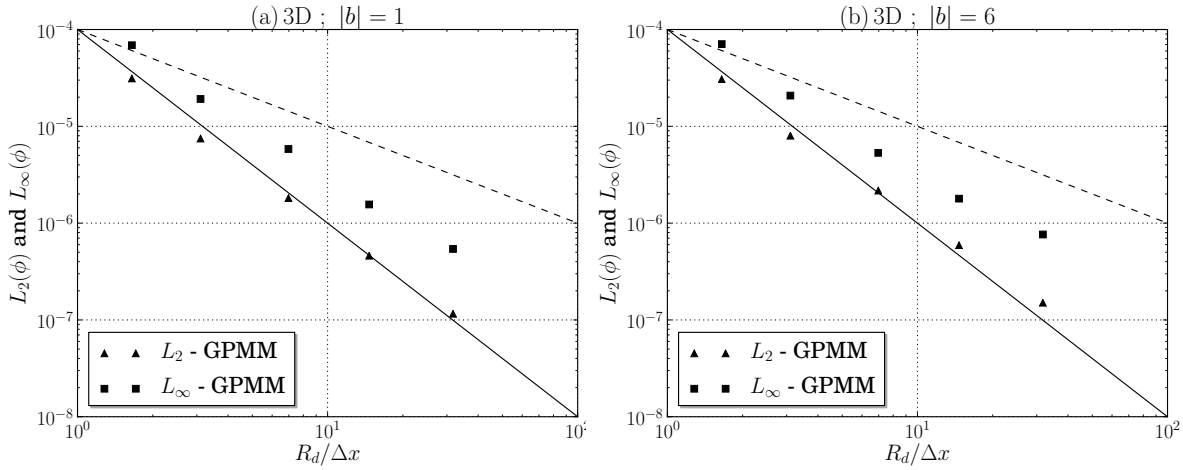


Figure 3.3: L_2 and L_∞ of the reconstructed signed-distance function ϕ with the 3D GPMM algorithm (Geometric-Projection Marker Method), in first (a) and sixth (b) narrow-band levels $|b|$. The dashed and solid lines are first and second-order convergence, respectively.

Table 3.1: Error norms of the reconstructed signed-distance function ϕ using the 3D GPMM algorithm, in first and sixth band levels for each resolution. The corresponding order of convergence are given within parentheses.

$R_d/\Delta x$	$L_2(\phi_1)$	$L_\infty(\phi_1)$	$L_2(\phi_6)$	$L_\infty(\phi_6)$
1.649	3.122×10^{-5}	6.880×10^{-5}	3.064×10^{-5}	7.065×10^{-5}
3.104	7.474×10^{-6} (2.26)	1.917×10^{-5} (2.02)	8.002×10^{-6} (2.12)	2.077×10^{-5} (1.94)
6.973	1.815×10^{-6} (1.75)	5.829×10^{-6} (1.47)	2.169×10^{-6} (1.61)	5.322×10^{-6} (1.68)
14.66	4.608×10^{-7} (1.84)	1.559×10^{-6} (1.77)	5.909×10^{-7} (1.75)	1.791×10^{-6} (1.47)
31.74	1.159×10^{-7} (1.79)	5.413×10^{-7} (1.37)	1.506×10^{-7} (1.77)	7.631×10^{-7} (1.10)

2D and 3D using the following error norms:

$$L_2(\kappa) = \frac{1}{\kappa_{\Gamma,\mathrm{th}}} \sqrt{\frac{1}{N_{\mathrm{cp}}} \sum_{ik=1}^{N_{\mathrm{cp}}} \left(\kappa_{\Gamma,\mathrm{th}} - \kappa_{\Gamma,ik}\right)^2} \quad \text{and} \quad L_\infty(\kappa) = \frac{1}{\kappa_{\Gamma,\mathrm{th}}} \max_{\mathrm{cp}} \left|\kappa_{\Gamma,\mathrm{th}} - \kappa_{\Gamma,ik}\right| \tag{3.2}$$

where N_{cp} stands for the total number of node pairs crossing the interface Γ, $\kappa_{\Gamma,\mathrm{th}}$ is the constant theoretical curvature of the circle/sphere, and $\kappa_{\Gamma,ik}$ is the pair-based interface curvature, for a node pair ik among the crossing pairs, obtained with Eq. (2.29). Results are shown in Fig. 3.4, for the two different discretizations of the Hessian operator $\boldsymbol{\mathcal{H}}\left(\phi_{\mathrm{GPMM}}\right)$ in Eq. (2.26). The non-compact discretization consists in applying two consecutive second-order gradients to the reconstructed signed-distance function ϕ_{GPMM} (Eq. (2.28)), leading to a non-compact stencil, whereas a smaller stencil is used in the compact formulation: only direct-neighbor points are used to compute the Hessian (Eq. (2.27)). Slight convergence of the L_2-norm is observed for the 2D and 3D interface curvatures computed using the non-compact Hessian matrix. However, using a compact stencil leads to divergence of the curvature for both L_2 and L_∞ norms, in 2D and 3D. Eventually, a comparison in 3D between the non-compact Goldman curvature and a non-compact discretization of the classical curvature $\kappa = \boldsymbol{\nabla} \cdot \mathbf{n}$ is shown in Fig. 3.5: better convergence properties of the non-compact Goldman curvature are displayed, despite slightly higher errors at very low resolutions. As a consequence, the non-compact Goldman formulation will be used in the following.

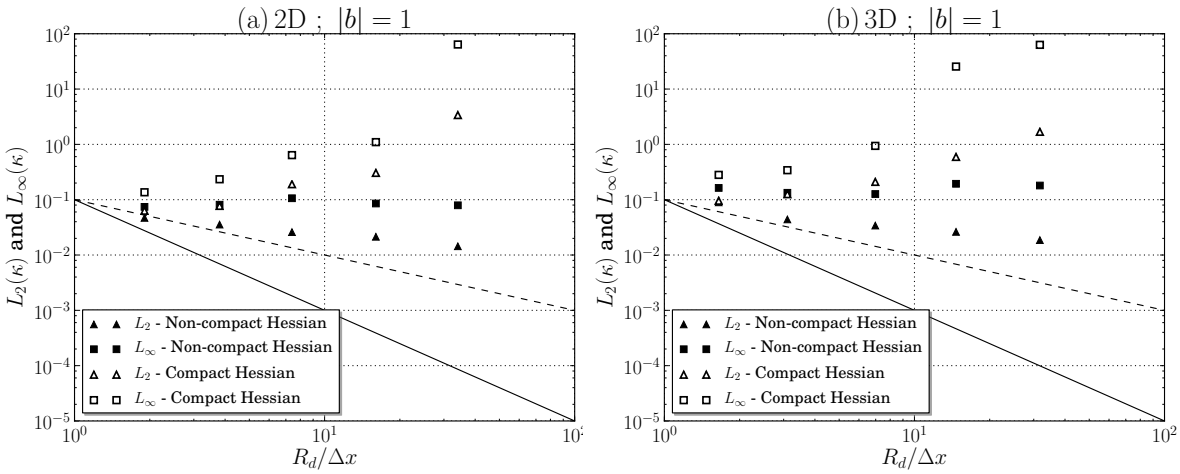


Figure 3.4: L_2 and L_∞ norms of the 2D (a) and 3D (b) interface curvature κ_Γ, considering two spatial discretizations of the Hessian operator: with compact and non-compact stencils. The dashed and solid lines are first and second-order convergence, respectively.

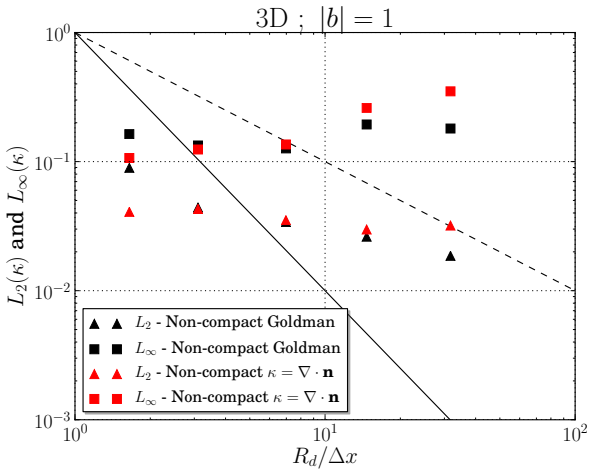


Figure 3.5: L_2 and L_∞ norms of the 3D interface curvature κ_Γ, using Goldman's formula and the classical formula, both discretized with non-compact stencils. The dashed and solid lines are first and second-order convergence, respectively.

3.2 Interface transport tests

In order to assess the accuracy, spatial convergence and robustness of the ACLS procedure, classical interface transport tests are performed.

3.2.1 Static sphere on static tetrahedral grids

A 3D static sphere is initialized on a homogeneous tetrahedral mesh. The initial signed-distance and ACLS functions are set as:

$$\phi_0(x,y,z) = R - \sqrt{x^2+y^2+z^2} \qquad \psi_0(x,y,z) = \frac{1}{2}\left(\tanh\left(\frac{\phi_0(x,y,z)}{2\varepsilon}\right)+1\right) \tag{3.3}$$

where R is the radius of the sphere. The two reinitialization equations Eq. (2.5) [31] and Eq. (2.7) [35] are compared at two grid resolutions: 6 and 12 points in the sphere radius R, and each of them is performed $50,000$ times throughout the runs. As homogeneous tetrahedra are used to tessellate the domain, the mesh cell size is computed as $\Delta x = \sqrt[3]{V_i}$, where V_i is the control volume based on the dual mesh around a node i. Results are presented in Fig. 3.6. As presented in [35] for

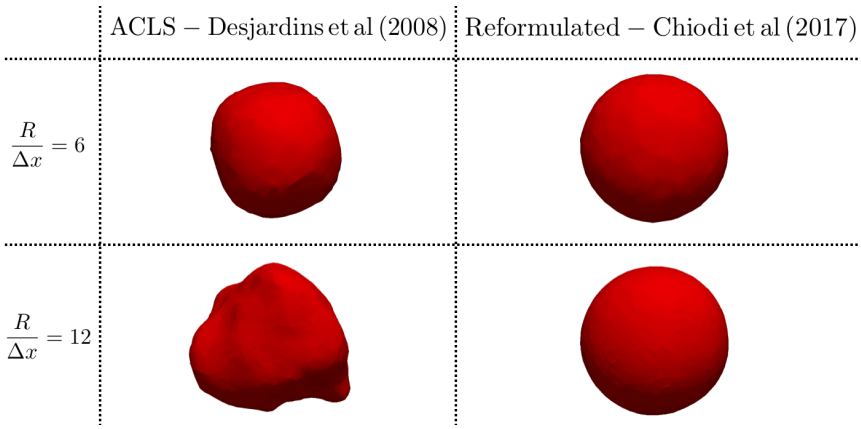


Figure 3.6: 3D static sphere transport test on two tetrahedral grids: comparison between Eq. (2.5) (original ACLS reinitialization) and Eq. (2.7) (reformulated reinitialization) after 50,000 iterations of reinitialization.

cartesian meshes, it is observed that the reformulated reinitialization Eq. (2.7) induces much less artificial deformation of the interface at high levels of reinitialization, here on tetrahedral meshes. Thus, the conclusion of [35] can be extended to unstructured meshes.

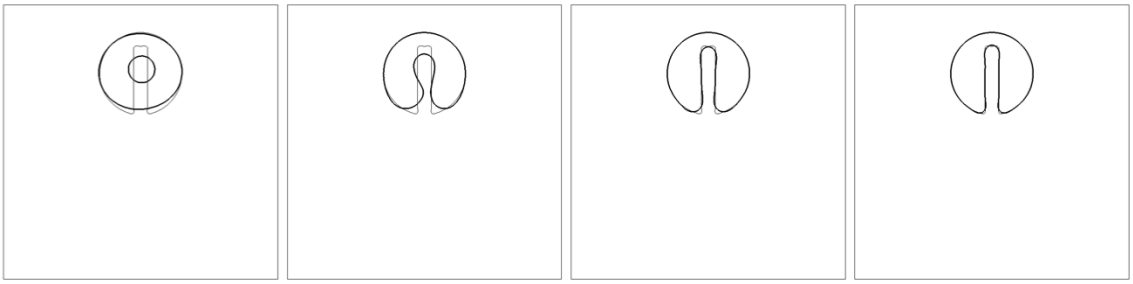
3.2.2 Rotation of a notched disk on static triangular grids

In order to assess the robustness of the methods presented above for the transport of interfaces with topology that includes sharp edges and corners, the classical notched disk rotation case, introduced by S.T. Zalesak in [173], is first run on homogeneous triangular grids. A two-dimensional disk of radius $R = 0.15$, with a notch of width $w = 0.05$, is initialized at $(x,y) = (0, 0.25)$ in a $[-0.5; 0.5]^2$ square. Given the unit angular velocity Ω and the origin of the domain as the center of rotation $(x_0, y_0) = (0,0)$, the velocity field is:

$$\mathbf{u}(x,y) = \begin{pmatrix} -\Omega(y-y_0) \\ \Omega(x-x_0) \end{pmatrix} = \begin{pmatrix} -y \\ x \end{pmatrix} \tag{3.4}$$

Thus, the disk has performed one full rotation about the origin when $t/T' = t\Omega/2\pi = 1$, with T' the time necessary for a single rotation. In the present paragraph, 10 rotations are simulated (i.e. up to $t/T = t/(10T') = 1$) on triangular meshes with a CFL of 0.48, which in theory should not change the shape of the slotted disk. Three reinitialization steps are done at each temporal iteration and $\varepsilon = \Delta x/2$. The narrow band is of width $|b|_{\max} = 12$ to ensure that $|\boldsymbol{\nabla}\psi| = 0$ at the edge of the band and thus better preserve mass.

Final interface contours after 10 rotations are presented in Fig. 3.7 for various mesh resolutions. Even if not mandatory as the exact solution is known to be the initial condition, the results are superimposed over a highly-resolved simulation (400 points in the disk radius), which is included in the convergence study thereafter. Good mesh convergence properties for interface dynamics are demonstrated: at the highest resolution shown (100 points in the disk radius), the initial shape is well retrieved. Fig. 3.8 shows two scalar fields, namely the signed narrow-band



(a) $\frac{R}{\Delta x} = 12.5$ (b) $\frac{R}{\Delta x} = 25$ (c) $\frac{R}{\Delta x} = 50$ (d) $\frac{R}{\Delta x} = 100$

Figure 3.7: Interface contours after 10 rotations for various mesh resolutions. The thin line represents a simulation performed with 400 points in the disk radius.

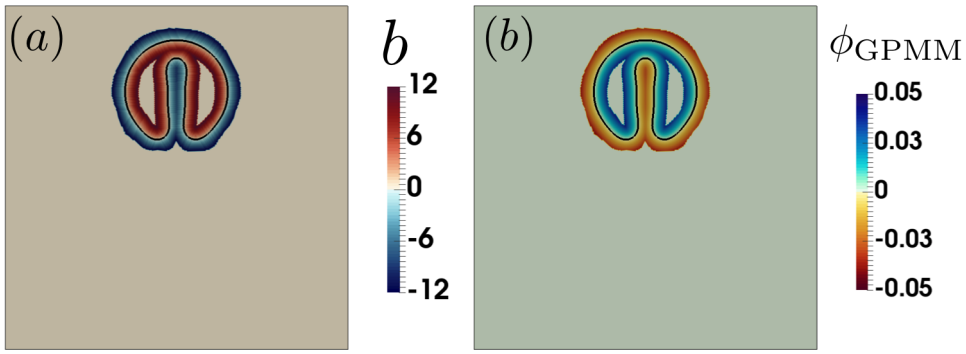


Figure 3.8: Scalar fields after 10 rotations with $R/\Delta x = 50$. (a) Signed narrow-band levels b; (b) Reconstructed signed-distance function ϕ_{GPMM}. Black contours represent the interface.

levels b and the reconstructed signed-distance function using the GPMM algorithm ϕ_{GPMM}, after 10 rotations for $R/\Delta x = 50$. These fields are seen to be smooth and well computed. Fig. 3.9 depicts the node normal vector field around the interface in the same computational conditions: as detailed in Section 2.2.2, computing the normals from a signed-distance function avoids spurious normal orientation.

To quantitatively assess the accuracy of the method, we use a methodology inspired by [137] to compute boundedness, shape, and mass errors. The boundedness error E_{bound} corresponds to the largest violation of bounds of the ACLS function ψ during the simulations ($0 \leq t \leq T$),

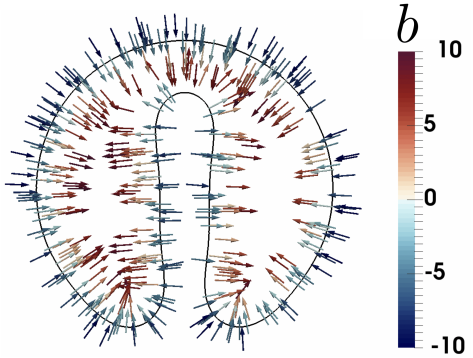


Figure 3.9: Node normal vector field $\mathbf{n}_i = \boldsymbol{\nabla}\phi_{\mathrm{GPMM}}\Big|_i^{4\mathrm{th}}$ around the interface after 10 rotations with $R/\Delta x = 50$, colored by the signed narrow-band levels b.

whereas the shape and mass errors are computed at the end of the runs ($t = T$):

$$E_{\mathrm{bound}} = \max_{t=0\ldots T}\left(\max\left(-\min_{i=1\ldots N_{\mathrm{cv}}}(\psi_i(t)V_i), \max_{i=1\ldots N_{\mathrm{cv}}}((\psi_i(t)-1)V_i)\right)\right) \tag{3.5}$$

$$E_{\mathrm{shape}}(T) = \sum_{i=1}^{N_{\mathrm{cv}}}\Big|\psi_i(T) - \psi_i(0)\Big|V_i \tag{3.6}$$

$$E_{\mathrm{mass}}(T) = \Big|\sum_{i=1}^{N_{\mathrm{cv}}}\psi_i(T)V_i - \sum_{i=1}^{N_{\mathrm{cv}}}\psi_i(0)V_i\Big| \quad \text{and} \quad \text{mass loss}\,[\%] = \frac{100\Big|\sum_{i=1}^{N_{\mathrm{cv}}}\psi_i(T)V_i - \sum_{i=1}^{N_{\mathrm{cv}}}\psi_i(0)V_i\Big|}{\sum_{i=1}^{N_{\mathrm{cv}}}\psi_i(0)V_i} \tag{3.7}$$

where N_{cv} is the total number of node-based control volumes in the domain. These errors allow the estimation of the largest overshoot/undershoot in the domain, the discrepancies in the final shape compared to the initial shape, and the mass loss throughout the run, respectively. The computational efficiency is measured through a reduced computational time RCT:

$$\mathrm{RCT} = \frac{\mathrm{WCT} \times N_{\mathrm{cores}}}{N_{\mathrm{iter}} \times N_{\mathrm{cv}}} \tag{3.8}$$

with WCT the Wall Clock Time, N_{cores} the number of cores, and N_{iter} the number of temporal iterations performed. The results are presented in Table 3.2.

Although machine accuracy is not reached, the level of errors in mass remains very low for all resolutions. Moreover, neither undershoot nor overshoot are present at the end of the simulation (10 rotations) for most of the resolutions: some appear during these runs, but with an extremely small amplitude as demonstrated by the maximum boundedness error values, and are naturally corrected over time as the interface topology becomes smoother. The shape error is plotted in Fig. 3.10: the method exhibits between first and second-order convergence on two-dimensional triangular meshes for this test case. The parallel efficiency of the conservative level set solver is demonstrated through the RCT values, which remain low for all calculations.

3.2.3 Circle deformation in a vortex on static triangular grids

A two-dimensional deformation of a disk in a vortex is then run [174]. This test is used to evaluate the behavior of the algorithms at resolution limit, which is useful for two-phase flow simulations,

Table 3.2: Error norms for each mesh resolution at the end of the notched disk test, and computational cost of the runs.

$R/\Delta x$	E_{bound}	$E_{\text{shape}}(T)$	$E_{\text{mass}}(T)$	mass loss [%]	N_{cores}	RCT [μs]
12.5	7.567×10^{-7}	1.360×10^{-2}	1.410×10^{-7}	2.406×10^{-4}	1	3.3
25	3.707×10^{-7}	7.544×10^{-3}	1.158×10^{-7}	1.986×10^{-4}	1	2.9
50	1.654×10^{-7}	2.226×10^{-3}	1.736×10^{-9}	2.980×10^{-6}	2	3.5
100	4.641×10^{-8}	8.427×10^{-4}	1.874×10^{-8}	3.219×10^{-5}	8	4.9
200	1.295×10^{-8}	3.434×10^{-4}	3.473×10^{-9}	5.965×10^{-6}	28	5.8
400	3.368×10^{-9}	1.224×10^{-4}	3.076×10^{-9}	5.284×10^{-6}	112	5.8

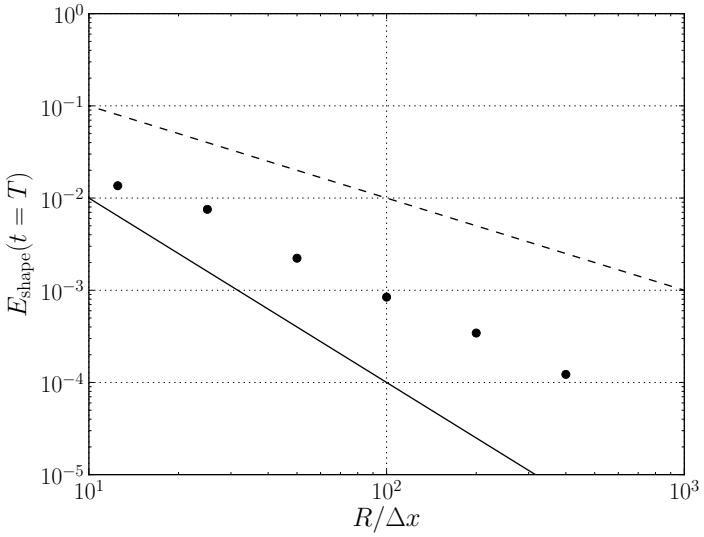


Figure 3.10: Shape error E_{shape} after 10 rotations of the 2D notched disk ($t = T$), for various mesh resolutions: from 12.5 to 400 points in the disk radius. The dashed and solid lines are first and second-order convergence, respectively.

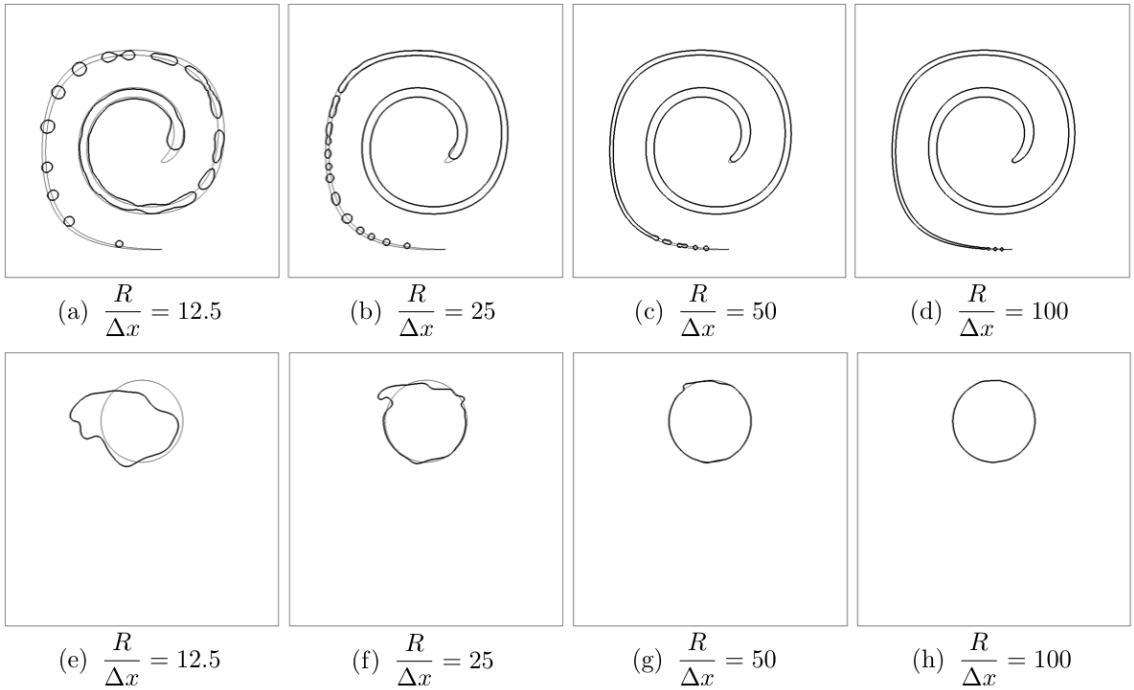
as thin ligaments are present. A disk of radius $R = 0.15$ is initially centered at $(x, y) = (0, 0.25)$ in a 2D box of dimensions $[-0.5; 0.5]^2$. The disk is deformed by a swirling velocity field:

$$\mathbf{u}(x, y, t) = \begin{pmatrix} -2\sin^2(\pi x)\sin(\pi y)\cos(\pi y)\cos(\pi t/T) \\ 2\sin^2(\pi y)\sin(\pi x)\cos(\pi x)\cos(\pi t/T) \end{pmatrix} \tag{3.9}$$

The disk is first stretched for $0 \leq t \leq T/2$, and then unstretched for $T/2 \leq t \leq T$. At $t = T$, the exact solution coincides with the initial condition. The total simulation time is chosen as $T = 8$. The initial CFL number is 0.64 for all performed simulations of this paragraph. Three reinitialization steps are done at each temporal iteration and $\varepsilon = \Delta x/2$. The narrow band is of width $|b|_{\max} = 12$.

Maximum and final interface deformation, at $t = T/2$ and $t = T$ respectively, are presented in Fig. 3.11 for various mesh resolutions. The results are superimposed over a highly-resolved simulation (400 points in the disk radius). Good mesh convergence properties are demonstrated: at the highest resolution shown (100 points in the disk radius), the original circular shape is perfectly retrieved. It should also be observed that at resolution limit, no locking phenomenon on the mesh

nor artificial surface tension occur: the tail breaks up into small circles. Indeed, in case of bad prediction of the resharpening term in the reinitialization Eq. (2.7), interfaces in ACLS/two-phase flow simulations can be locked on the grid and follow its pattern at resolution limit. As a consequence, the interfaces are kept from breaking up or merging: in such scenario, artificial surface tension occurs and the ACLS method is not able to correctly predict topology changes.



(a) $\frac{R}{\Delta x} = 12.5$ (b) $\frac{R}{\Delta x} = 25$ (c) $\frac{R}{\Delta x} = 50$ (d) $\frac{R}{\Delta x} = 100$

(e) $\frac{R}{\Delta x} = 12.5$ (f) $\frac{R}{\Delta x} = 25$ (g) $\frac{R}{\Delta x} = 50$ (h) $\frac{R}{\Delta x} = 100$

Figure 3.11: Interface contours at maximum deformation ($t = T/2$; top) and final shape ($t = T$; bottom) for various mesh resolutions. The thin line represents a simulation performed with 400 points in the disk radius.

Using the same methodology than in the previous section, boundedness, shape, and mass errors (Eq. (3.5) to (3.7)) are computed at the end of the runs ($t = T$). The results are presented in Table 3.3, with the reduced computational times of the runs (Eq. (3.8)). Again, the levels of error in bound and mass remain very low. Good mass conservation is ensured even at coarse grid resolutions. Convergence of the geometric error is displayed in Fig. 3.12: the method exhibits between first and second-order convergence on two-dimensional triangular meshes. Indeed, no numerical breakup occurs from $R/\Delta x = 200$, leading to different error types and thus different convergence rates. This behavior has been observed in other works, such as in [152].

3.2.4 Sphere deformation in a vortex on adaptive tetrahedral grids

The previous test case can be extended to 3D [174]. A sphere of radius $R = 0.15$ is initialized at $(x, y, z) = (-0.15, -0.15, -0.15)$ in a $[-0.5, 0.5]^3$ three-dimensional box. The advection velocity is specified as:

$$\mathbf{u}(x, y, z, t) = \begin{pmatrix} 2\sin^2(\pi x)\sin(2\pi y)\sin(2\pi z)\cos(\pi t/T) \\ -\sin(2\pi x)\sin^2(\pi y)\sin(2\pi z)\cos(\pi t/T) \\ -\sin(2\pi x)\sin(2\pi y)\sin^2(\pi z)\cos(\pi t/T) \end{pmatrix} \tag{3.10}$$

Table 3.3: Error norms at the end of the circle deformation test, and computational cost of the runs.

$R/\Delta x$	E_{bound}	$E_{\text{shape}}(T)$	$E_{\text{mass}}(T)$	mass loss [%]	N_{cores}	RCT [μs]
12.5	4.135×10^{-6}	3.642×10^{-2}	1.790×10^{-9}	2.518×10^{-6}	1	4.7
25	7.525×10^{-7}	8.452×10^{-3}	4.637×10^{-8}	6.551×10^{-5}	1	3.8
50	1.883×10^{-7}	1.970×10^{-3}	3.223×10^{-8}	4.558×10^{-5}	2	3.7
100	4.121×10^{-8}	6.016×10^{-4}	1.053×10^{-8}	1.490×10^{-5}	8	5.2
200	1.392×10^{-8}	2.719×10^{-4}	3.155×10^{-9}	4.464×10^{-6}	28	6.1
400	3.011×10^{-9}	1.289×10^{-4}	1.165×10^{-9}	1.649×10^{-6}	112	6.4

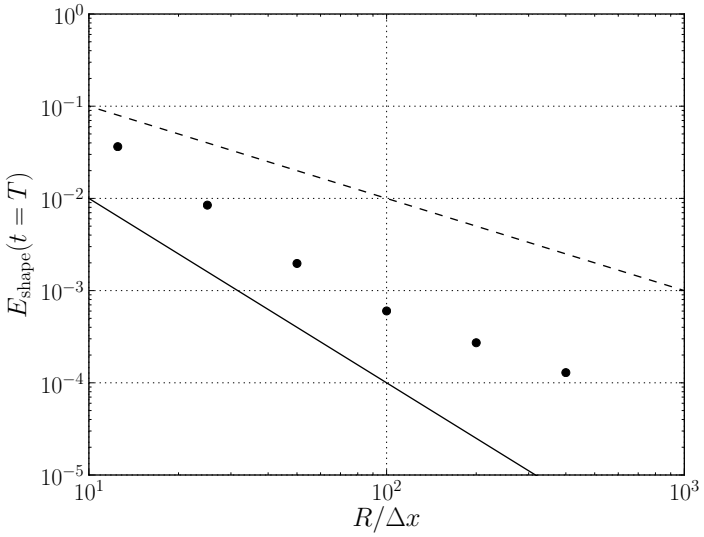


Figure 3.12: Shape error E_{shape} at the end of the 2D circle deformation in a vortex test ($t = T$), for various mesh resolutions: from 12.5 to 400 points in the disk radius. The dashed and solid lines are first and second-order convergence, respectively.

The simulation time is chosen as $T = 3$. At $t = T$, i.e. after one full stretch/unstretch cycle, the initial sphere is supposed to be recovered. At maximum deformation ($t = T/2$), a thin sheet is formed and breaks up numerically into small spheres if the mesh is not fine enough. Adaptive tetrahedral meshes are used for this test case, dynamically and locally refined during the runs around the interface location, with a metric $\Delta x_{\min}$. Here, it has been chosen to change the background mesh from one simulation to another so that $\Delta x_{\max}/\Delta x_{\min} = 5$ for all runs. Larger ratios will be considered in the next sections of this manuscript. For all calculations of this section, the refined region is of half-width $10\Delta x_{\min}$ ($N_p = 10$, see Fig. 2.8), and AMR is triggered each time the interface is closer than 5 cells from the border of this fine region. The narrow-band width $|b|_{\max}$ is set to 10, and the initial CFL number is 0.64.

Interface contours are seen in Fig. 3.13 and contours along with the tetrahedral cell-size distribution in Fig. 3.14: good convergence is observed, and the final shape is almost retrieved for $R/\Delta x_{\min} = 60$. The results are in accordance with the ones obtained for the two-dimensional case. The computational efficiency, measured through the reduced computational time RCT

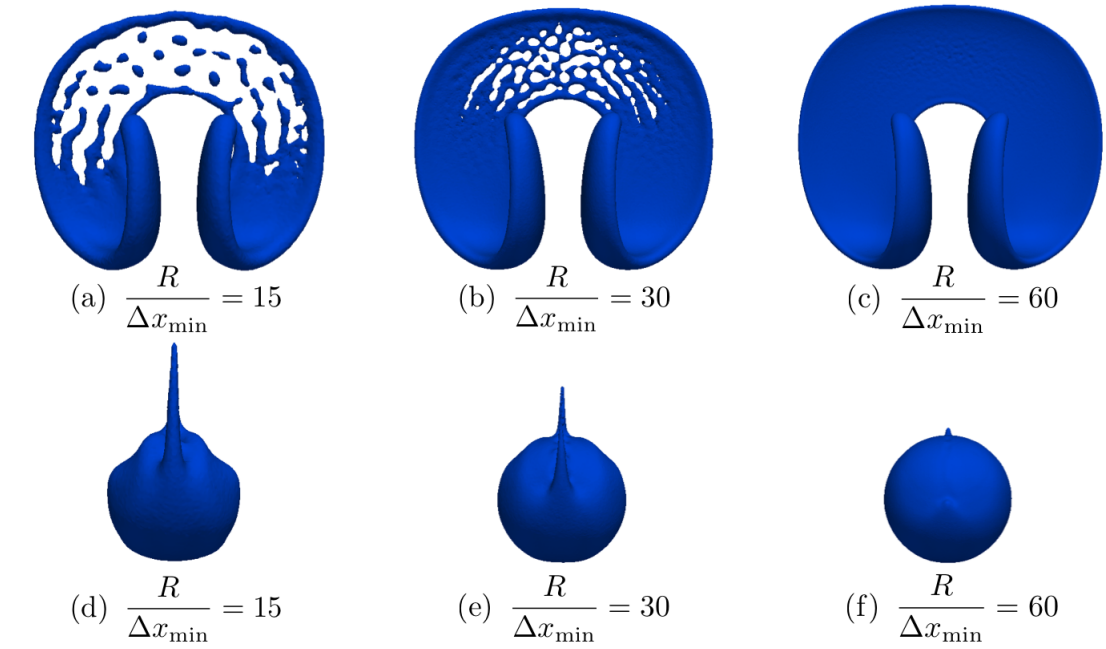


Figure 3.13: Interface contours at maximum stretch ($t = T/2$; top) and final shape ($t = T$; bottom) for various adaptive mesh resolutions.

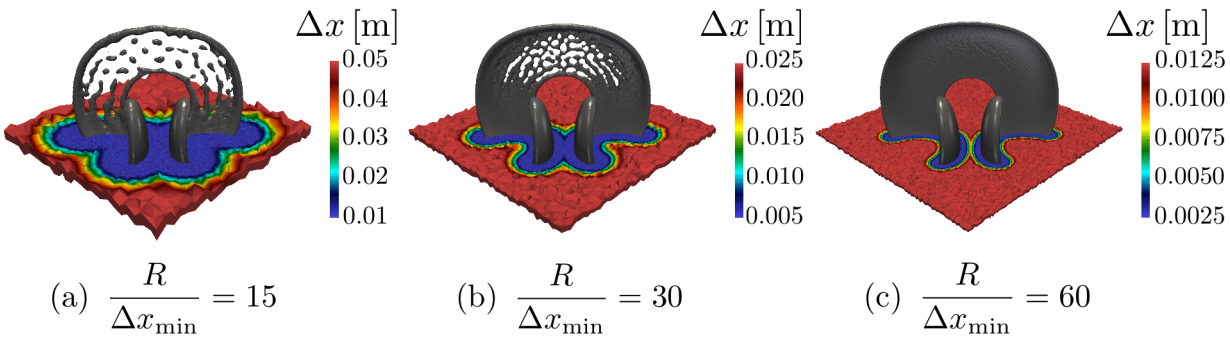


Figure 3.14: Tetrahedral cell size distribution at maximum stretch ($t = T/2$) for various adaptive mesh resolutions.

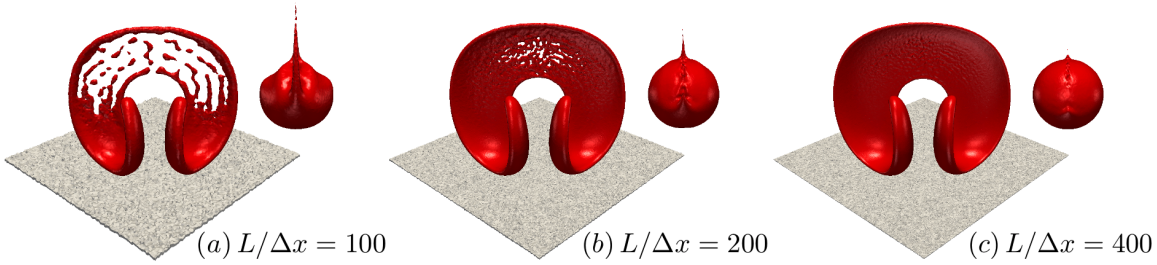
(Eq. (3.8)), is presented in Table 3.4. The RCT is computed for each of the two parts of the

Table 3.4: Computational costs of the sphere deformation simulations using adaptive tetrahedral grids

$R/\Delta x_{\min}$	N_{cores}	$N_{\text{cells,ini}}/N_{\text{cores}}$	RCT [μs]	RCT_{ACLS} [μs]	RCT_{AMR} [μs]	$\mathcal{N}_{\text{AMR,substeps}}/\mathcal{N}_{\text{AMR}}$	$N_{\text{cores}} \times$ WCT [h]
15	10	93,361	150.8	77.2	73.6	4.4	14.1
30	28	122,645	206.6	86.3	120.3	5.4	141.6
60	224	71,239	366.6	99.4	267.2	8.1	2271.1

procedure needed in this test case: the interface-capturing technique (ACLS) and the remeshing part (AMR). The different values of RCT presented in this table relate to each other as: $\text{RCT} = \text{RCT}_{\text{ACLS}} + \text{RCT}_{\text{AMR}}$. The increase in computational time remains low when increasing the resolution, and is mainly due to adaptive mesh refinement. Indeed, the AMR processes

triggered throughout the runs require more substeps to converge to a good-quality grid in average when increasing the resolution, due to the increasing number of bad quality cells at interfaces between processors. The average number of adaptation substeps necessary per grid adaptation process, $\mathcal{N}_{\mathrm{AMR,substeps}}/\mathcal{N}_{\mathrm{AMR}}$ in Table 3.4, increases with resolution, leading to an increase in $\mathrm{RCT}_{\mathrm{AMR}}$. Eventually, for an easy comparison with other results in the literature of level-sets, three additional simulations using uniform static tetrahedral grids are included, with increasing resolution from $L/\Delta x = 100$, with L the length of the box. The results are presented in Fig. 3.15 and Table 3.5. In this table, as uniform static grids have been used, RCT is the Reduced Computational Time of the pure ACLS interface-capturing technique (i.e. without AMR). Fig. 3.15 shows similar results compared to the study with AMR, for equivalent resolutions, while Table 3.5 demonstrates the great scalability of the pure ACLS interface-capturing procedure. The cost in CPU-hours is also included and can be compared to the results of Table 3.4: a gain of factor 10 in cost is displayed when using AMR for the finest simulation.



(a) $L/\Delta x = 100$ (b) $L/\Delta x = 200$ (c) $L/\Delta x = 400$

Figure 3.15: Interface contours at maximum stretch ($t = T/2$) and final shape ($t = T$) for various uniform mesh resolutions.

Table 3.5: Computational costs of the sphere deformation simulations using uniform tetrahedral grids

$L/\Delta x$	$R/\Delta x$	N_{cores}	$N_{\mathrm{cells}}/N_{\mathrm{cores}}$	RCT [μs]	$N_{\mathrm{cores}} \times$ WCT [h]
100	15	80	107,715	85.9	64.4
200	30	640	107,715	87.3	1281.8
400	60	5120	107,715	97.1	22186.7

3.3 Capillary-driven flows

3.3.1 Damping of a capillary wave on static triangular grids

The damping of a two-dimensional surface wave is then studied. Two superposed immiscible viscous fluids at rest are separated by a flat interface, initially perturbed by a small-amplitude sinusoidal capillary wave (see Fig. 3.16). The initial signed-distance function is written as:

$$\phi_0(x, y) = -y - A_0 \cos\left(\frac{2\pi x}{\lambda}\right) \tag{3.11}$$

in a two-dimensional $[-\lambda/2; \lambda/2]^2$ domain which is periodic in the x-direction, and with slip walls for top and bottom boundaries. The initial wave amplitude is chosen as $A_0 = \lambda/100$, with $\lambda = 2\pi$

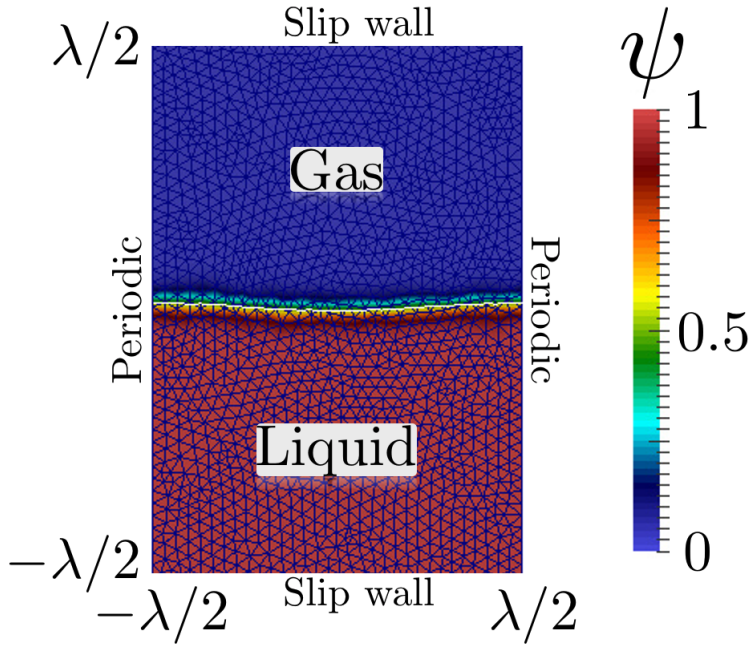


Figure 3.16: Illustration of the damped wave test case.

the wavelength of the perturbation. The initial perturbation is damped over time due to viscous dissipation.

The results are compared to analytical solutions from Prosperetti's theory [175], which quantifies the viscous damping for vanishingly small amplitudes and infinite domains as an initial-value problem. The theoretical solution for two fluids with equal kinematic viscosity ν and $\lambda = 2\pi$ is:

$$A_{\text{th}}(t) = \frac{4(1-4\beta)\nu^2}{8(1-4\beta)\nu^2+\omega_0^2} A_0 \text{erfc}\sqrt{\nu t} + \sum_{i=1}^{4} \frac{z_i}{Z_i} \left(\frac{A_0\omega_0^2}{z_i^2-\nu} \right) \exp\left((z_i^2-\nu)t\right) \text{erfc}\left(z_i\sqrt{t}\right) \tag{3.12}$$

where z_i are the four roots - two pairs of conjugated complex numbers - of the following fourth-order polynomial equation:

$$z^4 - 4\beta\sqrt{\nu}z^3 + 2(1-6\beta)\nu z^2 + 4(1-3\beta)\nu^{3/2}z + (1-4\beta)\nu^2 + \omega_0^2 = 0 \tag{3.13}$$

and $Z_i = \prod_{j=1,j\neq i}^{4}(z_j - z_i)$. In the case of equal density ρ, $\beta = 1/4$ and the normal-mode oscillation frequency ω_0 is computed from the dispersion relation as follows:

$$\omega_0^2 = \frac{\sigma}{2\rho} \tag{3.14}$$

The non-dimensional time is $t^* = \omega_0 t$. The calculations are performed on three homogeneous triangular grids ($\lambda/\Delta x = 30; 100; 200$) up to $T^* = \omega_0 T = 25$, which corresponds to approximately 4 oscillations. The unity density ratio case is investigated, gravity is neglected, and 3 steps of reinitialization are performed at each timestep. The physical and numerical parameters of the runs are summarized in Table 3.6.

The temporal evolution of the wave amplitude over time is depicted in Fig. 3.17 (a). The damping is correctly reproduced from 100 points in the wavelength λ in the case of equal densities. Fig. 3.17 (b) shows the instantaneous relative amplitude error $|A(t) - A_{\text{th}}(t)|/A_0$ and confirms

Table 3.6: Physical and numerical parameters for the viscous damping of a capillary wave case.

ρ_l/ρ_g	μ_l/μ_g	$\mathrm{La} = \rho\lambda\sigma/\mu^2$	$\lambda/\Delta x$	Δt^*
1	1	3000	30; 100; 200	0.001

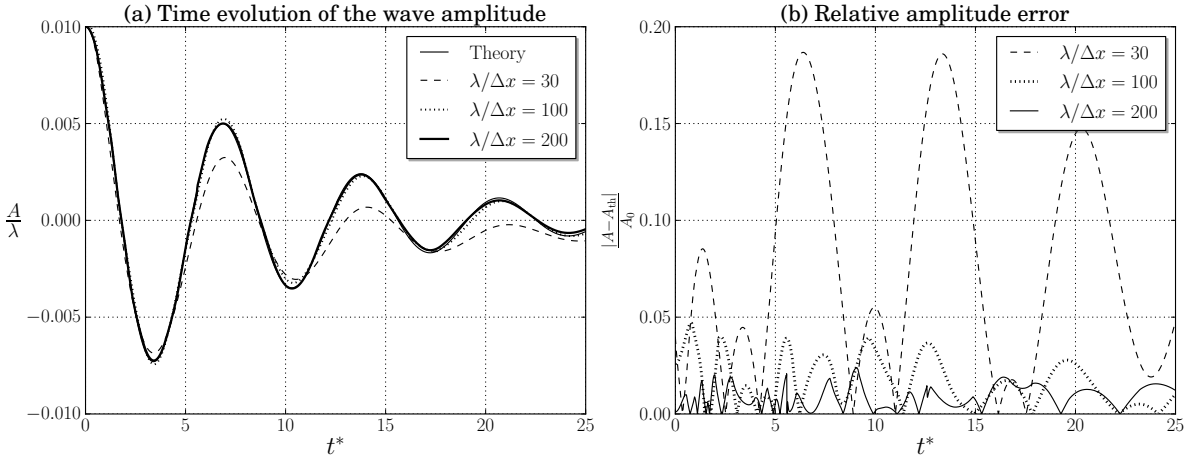


Figure 3.17: (a) Temporal evolution of the dimensionless wave amplitude A/λ over non-dimensional time $t^* = \omega_0 t$; (b) Instantaneous relative amplitude error $|A - A_{\mathrm{th}}|/A_0$.

the convergence properties. To quantify further the differences between theory and numerical simulations, the following L_2 norm is computed, as in [21]:

$$L_2 = \frac{1}{\lambda}\sqrt{\frac{1}{T^*}\int_{t^*=0}^{T^*} \left(A(t^*) - A_{\mathrm{th}}(t^*)\right)^2 \, \mathrm{d}t^*} \tag{3.15}$$

where A and A_{th} are the computed and theoretical wave amplitudes at a given instant, respectively. The L_2-norm values and the computational costs of the runs are presented in Table 3.7.

Table 3.7: Error levels of the L_2 norm (Eq. (3.15)) and reduced computational time (RCT; Eq. (3.8)) for each resolution. The corresponding order of convergence are given within parentheses.

$\lambda/\Delta x$	L_2	N_{cores}	$N_{\mathrm{cells}}/N_{\mathrm{cores}}$	RCT [μs]
30	9.602×10^{-4}	1	2028	23.2
100	2.042×10^{-4} (1.29)	1	22,476	13.3
200	1.035×10^{-4} (0.98)	4	22,520	20.2

The results show first-order convergence on triangular grids in the case of unity density ratio. In order to compare with other solvers for this well-known test case, the L_2 relative error norm, in which the numerical and theoretical amplitudes are normalized by the amplitude of the initial perturbation A_0, is displayed in Fig. 3.18 alongside the results from the CSF method [154] and the ones from Popinet 2009 [21], both obtained for cartesian grids. The latter method has proven to

be one of the most accurate methods available for cartesian grids. As mentioned in [21], the CSF results suffer from the influence of the top and bottom boundaries, as a squared domain has been used. This results in a much lower convergence at higher resolutions. A squared domain has also been used in our study, but the influence of the top and bottom boundaries seems less important. The first-order convergence, which is still satisfactory for unstructured grids, can be attributed to the weak convergence of the curvature computation on unstructured grids. In order to improve the accuracy and convergence, a more accurate surface tension model for unstructured grids will be needed in the future. Nonetheless, the overall algorithm robustness and physical meaning in the case of surface tension/viscosity interactions on unstructured grids is validated.

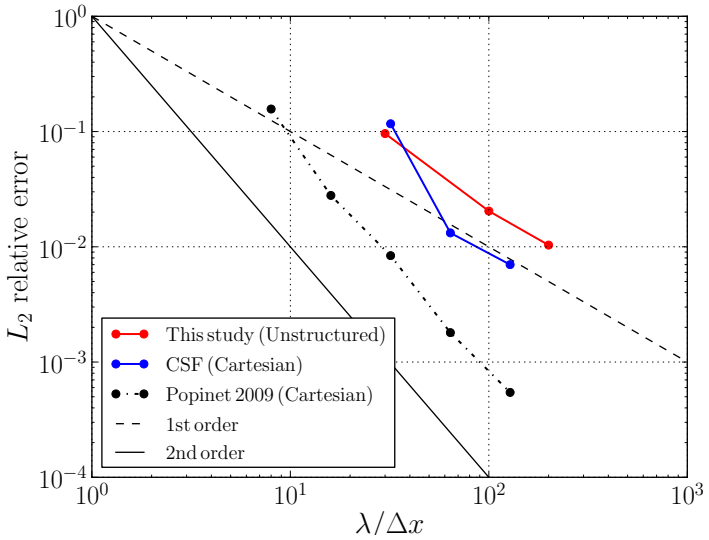


Figure 3.18: L_2 relative error norm for the present study, compared to the CSF method [154] and Popinet 2009 [21].

3.3.2 Spurious currents on static tetrahedral grids

This elementary test is used to assess the numerical errors induced by the surface tension computation procedure. For a static droplet, it aims to show the capability of a method to predict Laplace's law, i.e. the balance between surface tension and pressure forces. For a droplet in equilibrium, the theoretical velocity is 0 and thus non-physical velocities indicate the presence of spurious (or parasitic) currents. Exact balance (to machine accuracy) has been shown first on cartesian grids in [21]. More recently, Abu-Al-Saud et al. [52] demonstrated equilibrium to machine precision in a level set framework without reinitialization, if the simulation is run long enough relative to a dissipative timescale $T_\mu = \rho D^2/\mu$. As mentioned and demonstrated in [176] and [52], exact balance cannot be retrieved with level set methods when reinitialization is needed. Consequently, it is not possible to reach exact Laplace balance in our framework. Thus, in this work, we study spurious currents only to check the robustness and assess the numerical errors of the algorithms due to the initial interface perturbation/curvature and to high levels of reinitialization over a large number of solver iterations, in physical conditions close to the ones found in the range of the targeted practical applications, i.e. realistic liquid injection systems. To this mean, a high Laplace number is investigated: $\mathrm{La} = 2,000,000$.

To study the magnitude of spurious currents over time, the maximum Capillary number is

used: $\mathrm{Ca}_{\max} = \mu_l u_{\max}/\sigma$. Time is normalized using a visco-capillary timescale: $t^* = t/\tau$, with $\tau = \mu_l D/\sigma$. A three-dimensional droplet of diameter D is initialized in a closed unit centimetric square box. The simulations are performed on three static tetrahedral grids, up to $t^* = 10,000$. 3 reinitialization steps are performed at each iteration, and the explicit capillary time-step constraint STN in Eq. (2.61) is set to 0.71 for all resolutions. The physical and numerical parameters are presented in Table 3.8.

Table 3.8: Physical and numerical parameters for the measurements of spurious currents.

ρ_l/ρ_g	μ_l/μ_g	$\mathrm{La} = \rho_l \sigma D/\mu_l^2$	$D/\Delta x$
1000	50	2×10^6	10; 20; 40

Results are presented in Fig. 3.19 (a) and (b). Reasonably low errors ($\mathrm{Ca}_{\max} \sim 10^{-3}$) are seen for all resolutions, and a significant reduction of the spurious current magnitude is observed for $D/\Delta x = 40$. The spatial convergence of the spurious currents, which dominate the solution in such physical conditions, is displayed, and the presented procedure is demonstrated to be highly robust on unstructured grids.

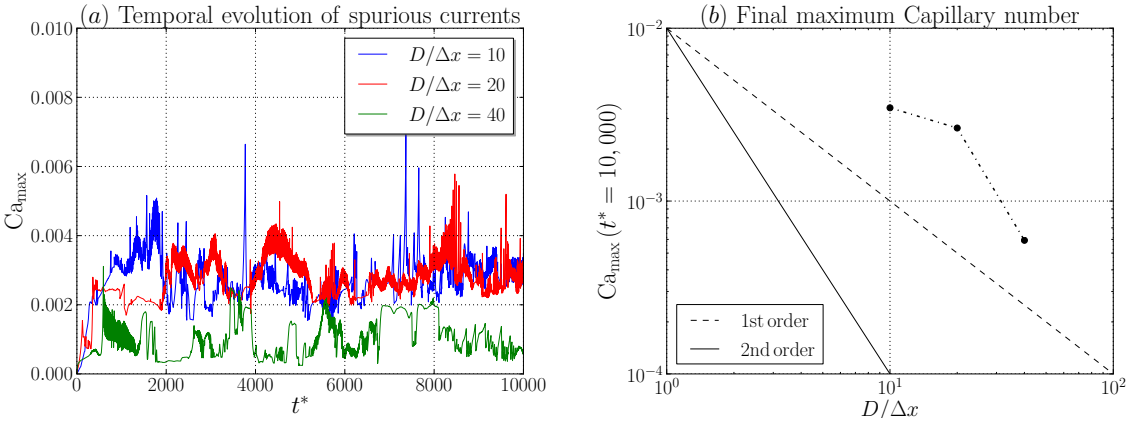


Figure 3.19: Spurious currents around a static droplet on tetrahedral meshes. $\mathrm{La} = 2,000,000$. $\rho_l/\rho_g = 1000$. $\mu_l/\mu_g = 50$.

3.3.3 Rayleigh-Plateau instability on static tetrahedral grids

In this section, following the studies of [21, 22, 31, 146], numerical predictions of the capillary instability are performed on 3D static unstructured meshes and validated against Weber's linear theory [177]. A static column of water of density ρ_l, dynamic viscosity μ_l, and surface tension σ in quiescent air is considered. This column is infinite in the x-direction and its radius is r_0. A small axisymmetric perturbation wave is imposed at the column surface. The perturbation is given by:

$$r(x,t) = r_0 + \varepsilon r_0 \exp(\omega t + ikx) \tag{3.16}$$

where ε is the perturbation, assumed to be small, $k = 2\pi/\lambda$ is the wavenumber, and ω is the growth rate. Using the non-dimensional wavenumber $\xi = kr_0$ and growth rate $\omega_0 = \sqrt{\sigma/(2\rho_l r_0^3)}$, the

equation for the growth rate in Weber's theory applied to static jets, for $\xi < 1$, is given by:

$$\omega^2 + \frac{3\mu_l \xi^2}{\rho_l r_0^2}\omega = \omega_0^2 \xi^2 (1 - \xi^2) \tag{3.17}$$

The mathematical developments leading to this result can be found in [178].

The simulations are performed in a 3D, periodic in the x-direction, cylindrical domain of length $[0; \lambda]$ and radius $[0; 3r_0]$. The signed-distance function is initialized as:

$$\phi_0(x, y, z) = r_0 - \varepsilon r_0 \cos(kx) - \sqrt{y^2 + z^2} \tag{3.18}$$

where $r_0 = (1/3) \times 10^{-3}\mathrm{m}$, and the initial velocity is set to 0. In order to keep the interface free from spurious signals, the initial perturbation is set to be of the same order of magnitude than the homogeneous tetrahedral-cell size: $\varepsilon r_0 \sim \Delta x$, here $\varepsilon = 0.06$. The runs are performed at various ξ values, ranging between 0 and 1 (unstable oscillatory solutions), and using a coarse effective resolution of $r_0/\Delta x = 6$. The definition of the cases can be found in Table 3.9. For all cases, the

Table 3.9: Cases definition for the Rayleigh-Plateau instability simulations.

Case	$\lambda\,[\mathrm{mm}]$	λ/r_0	$\xi = kr_0$	N_{cells}
1	16	48	0.13	2.5M
2	8	24	0.26	1.3M
3	4	12	0.52	637K
4	3.3	10	0.63	525K
5	2.8	8.5	0.74	453K
6	2.3	7	0.90	372K

narrow-band width is $|b|_{\max} = 12$ and 3 steps of reinitialization are performed at each timestep. Classical water/air parameters are used, and are summarized in Table 3.10. The Laplace number is $\mathrm{La} = \rho_l r_0 \sigma / \mu_l^2 = 18,530$.

Table 3.10: Physical parameters for the Rayleigh-Plateau instability simulations.

ρ_l/ρ_g	μ_l/μ_g	La
829	62	18,530

The interface contours for $\xi = 0.52$ (Case 3) can be seen in Fig. 3.20: implicit interface topology changes are correctly predicted. A qualitative comparison of the interface topology close to breakup between the present coarse simulation (Fig. 3.20 (c)) and a more resolved simulation: another static tetrahedral grid with $r_0/\Delta x = 24$, is shown in Fig. 3.21. The relative deformation based on the maximum radius is defined at $x = \lambda/2$ as:

$$\frac{r_{\max} - r_0}{\varepsilon r_0} = \exp(\omega t) \tag{3.19}$$

The characteristic time is $t_\sigma = 1/\omega_0 = 1.013\mathrm{ms}$. Using t_σ, Eq. (3.19) becomes:

$$\ln\left(\frac{r_{\max} - r_0}{\varepsilon r_0}\right) = \frac{\omega}{\omega_0}\frac{t}{t_\sigma} \tag{3.20}$$

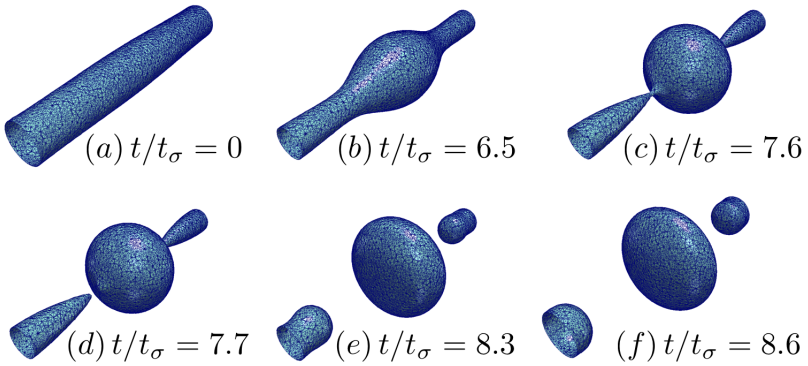


Figure 3.20: Rayleigh-Plateau instability on a homogeneous tetrahedral mesh: interface contours at various non-dimensional times t/t_σ. La $= 18,530$. $\xi = kr_0 = 0.52$. $r_0/\Delta x = 6$. $t_\sigma = 1/\omega_0$.

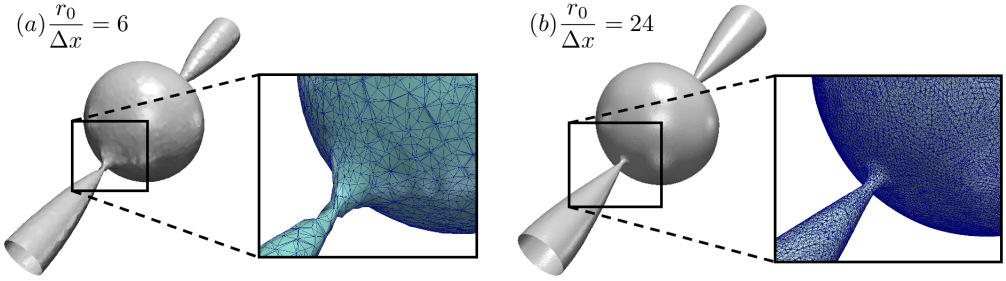


Figure 3.21: Interface contour in the pinch-off region. La $= 18,530$. $\xi = kr_0 = 0.52$. (a) $r_0/\Delta x = 6$; (b) $r_0/\Delta x = 24$.

This term is linear in time and the derivative is the non-dimensional growth rate. However, due to the use of fully 3D tetrahedral meshes, several causes of inaccuracy might deteriorate the quality of the computed growth rate: (i) the jet can be slightly off-centered in the radial direction, (ii) the maximum amplitude can be strongly shifted from the middle of the jet in the longitudinal direction. While problem (ii) is solved by choosing an initial perturbation of the same order of magnitude than the cell size, problem (i) requires a dedicated methodology. To post-process the results, the jet is discretized into rings of chosen thickness $\delta = 80\mu$m, and area A. For each ring i of each temporal solution n, the mean radius is computed from the mean center:

$$\overline{r}\Big|_i^n = \frac{1}{A}\int_A \sqrt{(y-\overline{y})^2 + (z-\overline{z})^2}\,\mathrm{d}A \tag{3.21}$$

where $\overline{y} = (1/A)\int_A y\,\mathrm{d}A$ and $\overline{z} = (1/A)\int_A z\,\mathrm{d}A$. The maximum mean radius $\overline{r}\Big|_{\max}^n$ is then computed for each temporal solution n. The temporal evolution of the relative deformation for $\xi = 0.52$ (Case 3) is plotted in Fig. 3.22 (a). Despite a non-linear phase at the beginning of the calculation, remaining until the disturbance has grown quite large, the non-dimensional growth rate ω/ω_0 is correctly predicted. To quantify the growth rate only in the linear part, a temporal mean is performed from $t/t_\sigma = 4$ to the breakup. The standard deviation SD of the growth-rate signal is also computed within the range considered:

$$\mathrm{SD} = \sqrt{\frac{1}{N-1}\sum_{n=1}^{N-1}\left(\frac{\omega}{\omega_0}\Big|_n - \frac{\omega}{\omega_0}\Big|_{\mathrm{mean}}\right)^2} \tag{3.22}$$

where N is the total number of temporal solutions.

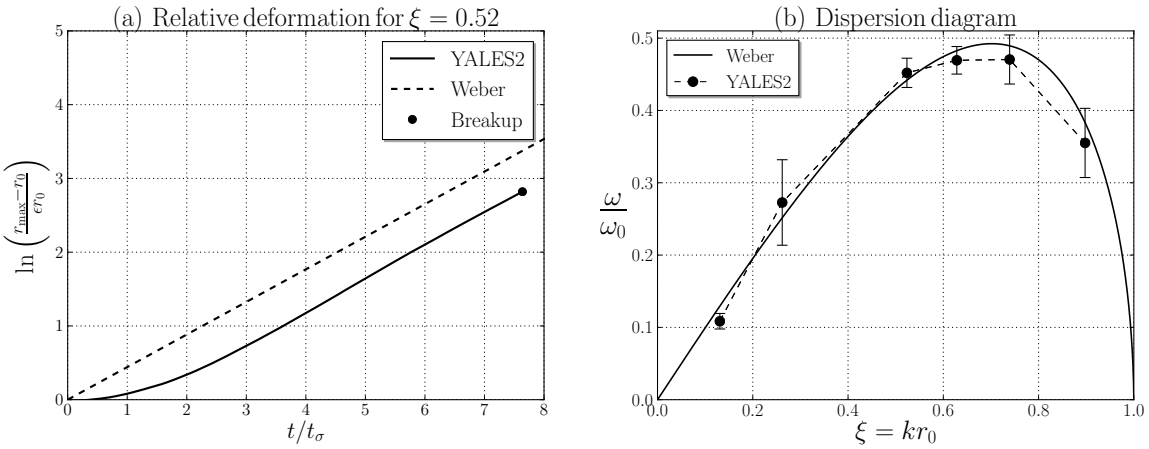


Figure 3.22: Rayleigh-Plateau instability. $\mathrm{La} = 18,530$. $r_0/\Delta x = 6$. (a) Temporal evolution of the relative deformation based on the maximum radius for $\xi = kr_0 = 0.52$, $t_\sigma = 1/\omega_0$. (b) Dispersion diagram; the error bars represent the standard deviation SD of the growth-rate signal $\pm$SD (see Eq. (3.22)).

The theoretical and numerical growth rates for all computed non-dimensional wavenumbers ξ are presented in Table 3.11 and depicted in the dispersion diagram of Fig. 3.22 (b). Larger standard deviations are obtained in the second and sixth cases ($\xi = 0.26$ and $\xi = 0.90$ respectively): the linear regime is more oscillatory, less clear to identify, in these two conducted simulations.

Table 3.11: Numerically predicted mean growth rates versus theoretical growth rates from Weber's theory, with $r_0/\Delta x = 6$.

Case	ξ	$\omega/\omega_0\|_{\mathrm{mean}}$	$\omega/\omega_0\|_{\mathrm{th}}$	Relative error	Standard deviation SD
1	0.13	0.109	0.130	-16.2%	0.011
2	0.26	0.273	0.252	+8.3%	0.059
3	0.52	0.452	0.442	+2.3%	0.020
4	0.63	0.469	0.483	-2.9%	0.019
5	0.74	0.470	0.489	-3.9%	0.034
6	0.90	0.355	0.383	-7.3%	0.048

3.3.4 3D droplet collision using adaptive mesh refinement

This section aims to validate the unstructured discretization of the reformulated reinitialization equation in the case of an interface merging scenario. This is a non-trivial case: a bad prediction of the resharpening term leads to the locking of the interface on the grid, and as a consequence keeps the two interfaces of the two droplets from merging. In order to demonstrate that the presented procedure allows implicit topology changes, a head-on collision of two equal-size droplets of water in air is computed.

The physical parameters involved in this process are the Weber number $\mathrm{We} = \rho_l u_r^2 D/\sigma$, with u_r the relative velocity of the two droplets; the Ohnesorge number $\mathrm{Oh} = \mu_l/\sqrt{\sigma\rho_l D}$; the non-dimensional impact parameter X, which is 0 for a head-on collision; and the droplet diameter

ratio Δ. The values are presented in Table 3.12 and correspond to the experiment of [179], in the reflexive separation regime. The narrow-band width is $|b|_{\max} = 10$, and the minimum cell size $\Delta x_{\min} = 6\mu$m is imposed in a refined region of half-width $10\Delta x_{\min}$ ($N_p = 10$, see Fig. 2.8). The mesh adaptation process is triggered automatically when the interface is 5 cells away from the border of the refined region, which leads to massive computational savings. Three steps of reinitialization are performed at each iteration.

Instantaneous interface positions and meshes are presented in Fig. 3.23, and show good interface quality and accuracy, even after a very large number of iterations ($\Delta t \sim 0.01\mu$s). Despite the presence of an entrapped bubble, which may be due to the lack of perfect symmetry with the unstructured grid, implicit topology changes are shown, and fair overall dynamics is displayed compared to the experimental results of [179], as depicted in Fig. 3.24. The computational efficiency is presented in Table 3.13.

Table 3.12: Physical parameters for the water/air head-on droplet collision simulation.

We	Oh	Δ	X
23	0.0047	1	0

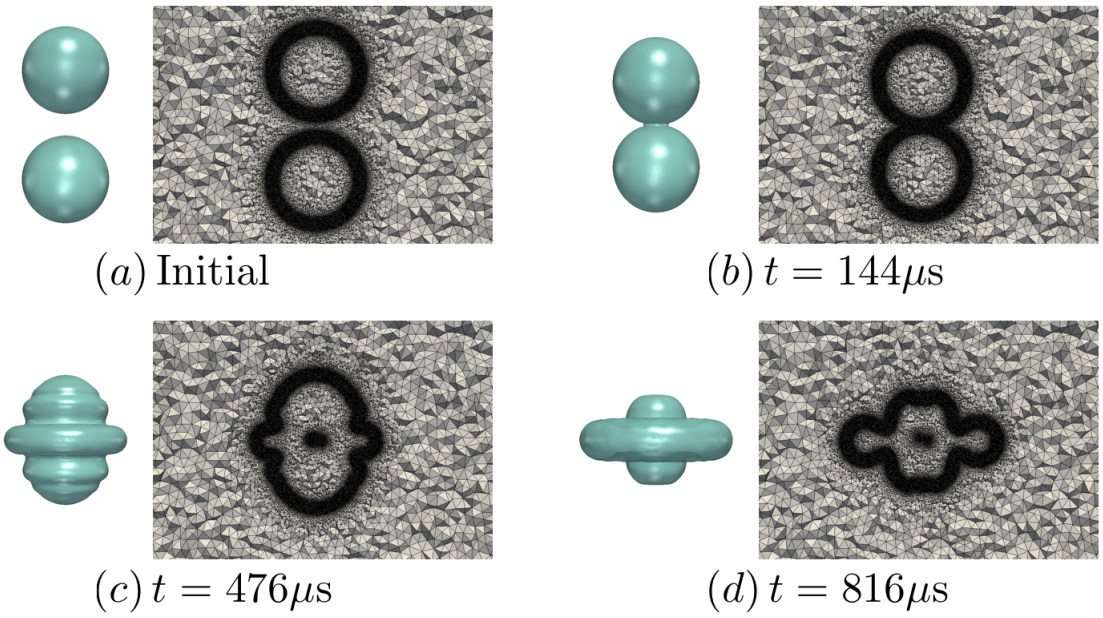


Figure 3.23: Water/Air droplet collision. We $= 23$; Oh $= 0.0047$; X=0; $\Delta = 1$; $D/\Delta x_{\min} = 133$.

Table 3.13: Computational performances of the water/air head-on droplet collision simulation.

$D/\Delta x_{\min}$	$\Delta x_{\max}/\Delta x_{\min}$	$N_{\text{cells,ini}}$	N_{cores}	RCT_{AMR} [μs]	RCT [μs]
133	17	36M	448	23.8	239.4

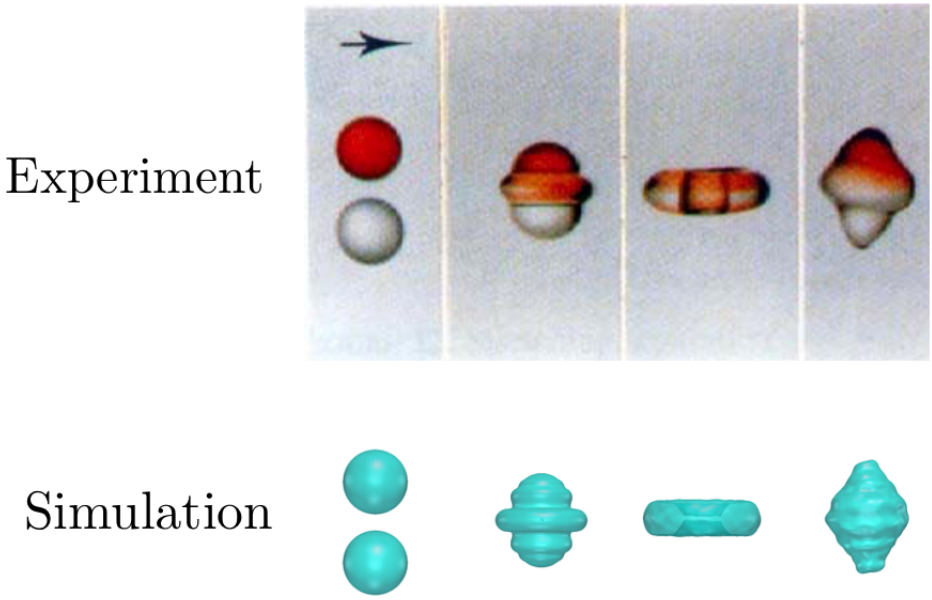


Figure 3.24: Water/Air droplet collision simulation compared to the experiment of [179], performed in the same physical conditions. We $= 23$; Oh $= 0.0047$; X=0; $\Delta = 1$; $D/\Delta x_{\min} = 133$.

CHAPTER 4

Application to injection systems

Disclaimer: Part of this chapter has been published in *Journal of Computational Physics* [216].

Contents

4.1 LES of a low-pressure water jet from a compound nozzle 99
4.1.1 Internal flow 100
4.1.2 External flow 102
4.1.3 Computational performances 102
4.2 LES of a high-pressure kerosene jet in crossflow 106
4.2.1 Numerical setup 106
4.2.2 Jet topology and breakup 108
4.2.3 Vertical trajectory and experimental validation 108
4.2.4 Computational performances 112

This fourth chapter aims to assess the capabilities of the YALES2 two-phase flow solver to perform massively-parallel high-fidelity LES simulations of complex turbulent atomizing liquid jets, using adaptive tetrahedral grids. Two different physical situations are investigated, both with high-density ratios: (i) A low-pressure water jet discharged from a triple-disk nozzle in quiescent air; (ii) A high-pressure kerosene jet atomized by a strong air crossflow. While atomization in case (i) is mainly governed by capillary instabilities and turbulence, case (ii) involves air-assisted atomization, driven by aerodynamic inertia. The computational costs and performances of the simulations are analyzed, as well as the overall behaviors of the jets, validated against experiments performed in the same physical conditions.

4.1 LES of a low-pressure water jet from a compound nozzle

This study follows the experiments of [39] and the simulations of [40]. A single low-pressure compound nozzle is considered: three shifted cylinders are superimposed, as depicted in Fig. 4.1, from which water is injected in quiescent air at Reynolds number $\mathrm{Re} = 3653$, based on the discharge orifice diameter $d_{\mathrm{inj}} = 180\mu\mathrm{m}$, and liquid Weber number $\mathrm{We_l} = 1061$. The shift of the disks induces non-axial velocity components, and a double-vortex flow is formed, as illustrated in Fig. 4.2.

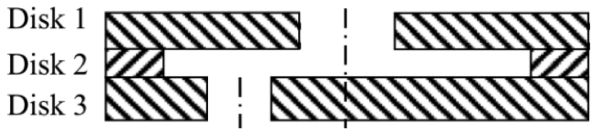


Figure 4.1: Compound nozzle geometry: triple-disk injector.

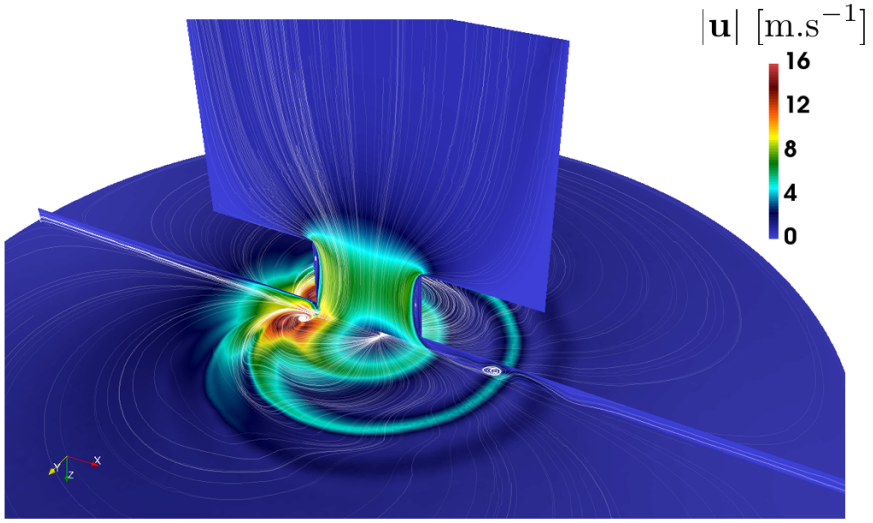


Figure 4.2: Illustration of the double-vortex internal flow in the cavity disk (Disk 2 in Fig. 4.1): velocity magnitude along with streamlines.

In these physical conditions, primary atomization is mainly driven by capillary effects such as ligament breakup and jet forming by bubble burst, and enhanced by liquid turbulence. The issued water jet is slow, and hence aerodynamic forces are negligible. As the injector geometry strongly conditions the primary breakup process, the unsteady resolution of the internal flow is important, as mentioned in [40].

In this work, the whole flow is computed by the mean of one Large-Eddy Simulation (LES), from the internal flow to the primary atomization process. The LES turbulence model is the dynamic Smagorinsky model [70, 71]. Adaptive tetrahedral meshes are used and two resolutions are investigated: $\Delta x_{\min} = 5\mu\text{m}$ and $\Delta x_{\min} = 2.5\mu\text{m}$. For the two simulations, the refined region is of half-width $10\Delta x_{\min}$ ($N_p = 10$, see Fig. 2.8), and AMR is triggered each time the interface is 5 cells away from the edge of this region. The jet morphology and angle are studied, and compared to the ones of [40].

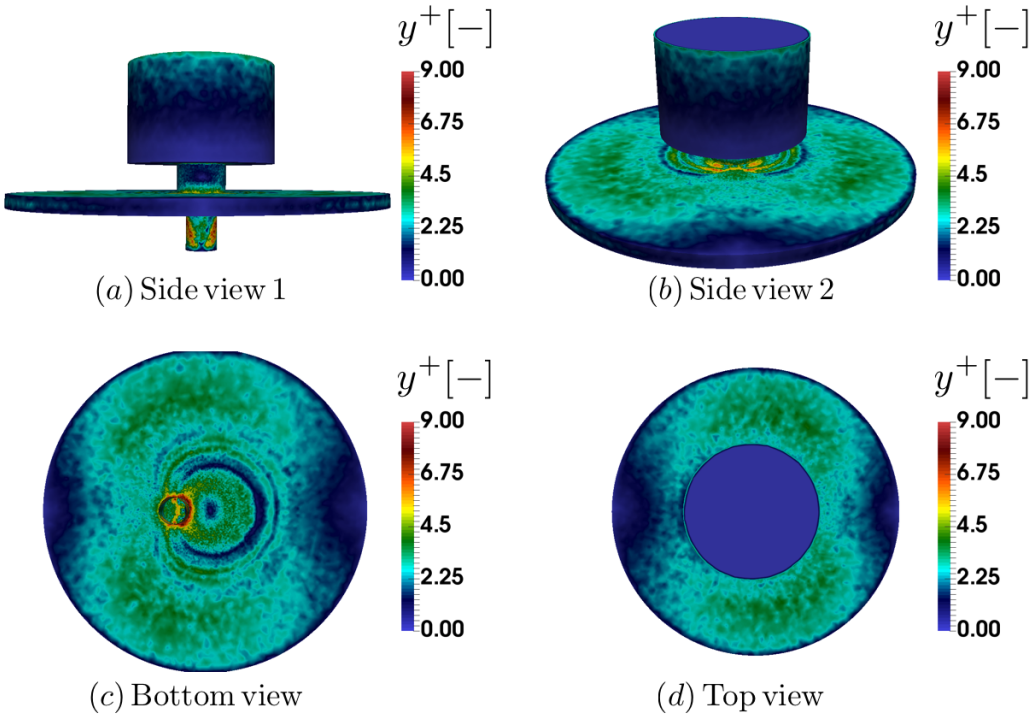
4.1.1 Internal flow

As presented above, the adaptive mesh refinement strategy is based on interface displacement, and thus the mesh size inside the injector, where only liquid is present, is the same for the two conducted calculations: the y^+ field represented at the walls is shown in Fig. 4.3. Reynolds temporal statistics of the velocity field (mean and root mean square) are accumulated using a weighted sampling at every time step as follows:

$$\overline{\mathbf{u}}\left(T_{\text{acc}}+\Delta t\right)=\frac{T_{\text{acc}}}{T_{\text{acc}}+\Delta t}\overline{\mathbf{u}}\left(T_{\text{acc}}\right)+\frac{\Delta t}{T_{\text{acc}}+\Delta t}\mathbf{u}\left(T_{\text{acc}}+\Delta t\right) \tag{4.1}$$

$$\mathbf{u}'\left(T_{\text{acc}}+\Delta t\right)=\sqrt{\frac{T_{\text{acc}}}{T_{\text{acc}}+\Delta t}\left(\mathbf{u}'^2\left(T_{\text{acc}}\right)+\overline{\mathbf{u}}^2\left(T_{\text{acc}}\right)\right)+\frac{\Delta t}{T_{\text{acc}}+\Delta t}\mathbf{u}^2\left(T_{\text{acc}}+\Delta t\right)-\overline{\mathbf{u}}^2\left(T_{\text{acc}}+\Delta t\right)} \tag{4.2}$$

with T_{acc} the total accumulation time. Converged temporal statistics of the flow velocity projected in the nozzle exit plane are shown in Fig. 4.4: the double-vortex secondary flow of the discharged disk (Disk 3 of Fig. 4.1) is clearly displayed.



(a) Side view 1 (b) Side view 2

(c) Bottom view (d) Top view

Figure 4.3: y^+ distribution represented at boundaries. $d_{\text{inj}} = 180\mu\text{m}$. $\text{Re} = 3653$. $\text{We}_\text{l} = 1061$.

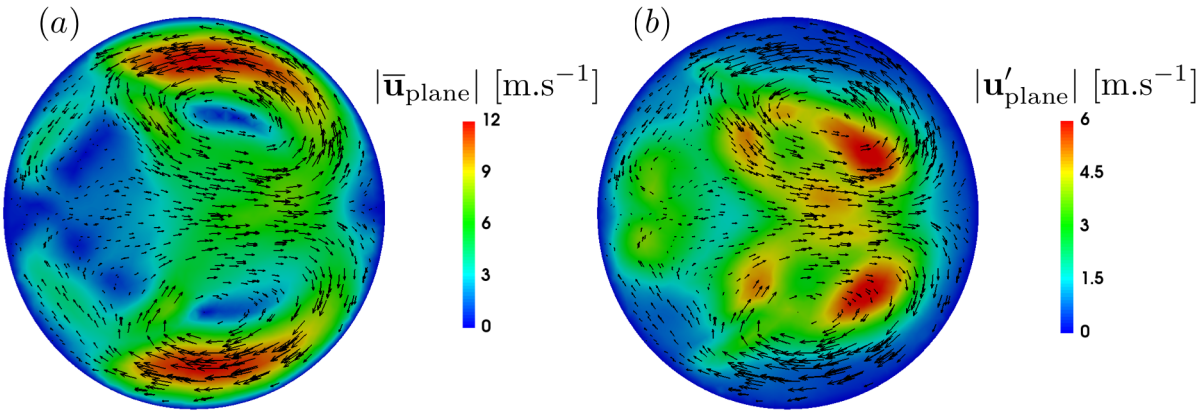


Figure 4.4: Velocity statistics: (a) Mean (Eq. (4.1)) and (b) Root Mean Square (Eq. (4.2)) velocity projected in the exit section. $d_{\text{inj}} = 180\mu\text{m}$. $\text{Re} = 3653$.

4.1.2 External flow

A comparison of interface topologies and vorticity-dominated regions at the two simulated resolutions are shown in Fig. 4.5. A clear difference is seen in the appearance of jet instabilities: a closeup on some capillary effects, especially the Rayleigh-Plateau instability is shown for the $2.5\mu\mathrm{m}$ case in Fig. 4.6. It depicts the temporal evolution of a ligament, circled in red in Fig. 4.6 (a), which is pinched off in Fig. 4.6 (b) until breakup (c) to produce two droplets. These outcome drops later collide and merge. These observations are simply not accessible with $\Delta x_{\min} = 5\mu\mathrm{m}$. The jet topology in the $2.5\mu\mathrm{m}$ case is close to the one obtained by a DNS simulation at $1.44\mu\mathrm{m}$, performed with a symmetry plane in [40] and to the experiment: the comparison is displayed in Fig. 4.7. This similarity is encouraging to target an even higher resolution. Fig. 4.8 shows the mesh cell-size distribution for both computed resolutions. The background mesh is kept identical for both, and it does not affect the quality of the solution: only smooth cell-size gradients are present in the domain, even with a higher cell-size ratio $\Delta x_{\max}/\Delta x_{\min} = 22.8$ in the $2.5\mu\mathrm{m}$ case.

Eventually, the spray angle α is estimated at various distances z/d_{inj} away from the nozzle exit for $d_{\mathrm{inj}}/\Delta x_{\min} = 72$, in order to compare with the simulations and experiments of [40]. An image processing technique has been employed on 250 frames, corresponding to $250\mu\mathrm{s}$ of physical time, of the projected shadow of the spray to obtain a single mean image: Fig. 4.9 (a). The angle values are presented in Fig. 4.9 (b): the angles are rather overestimated, especially close to the nozzle, compared to the experimental values presented in [40], which lie between 20° and 25°. This could be explained by the lack of atomization close to the nozzle in our numerical simulations. Nevertheless the overall behavior is correct, despite a more significant decrease of the angle with the distance from the nozzle exit.

4.1.3 Computational performances

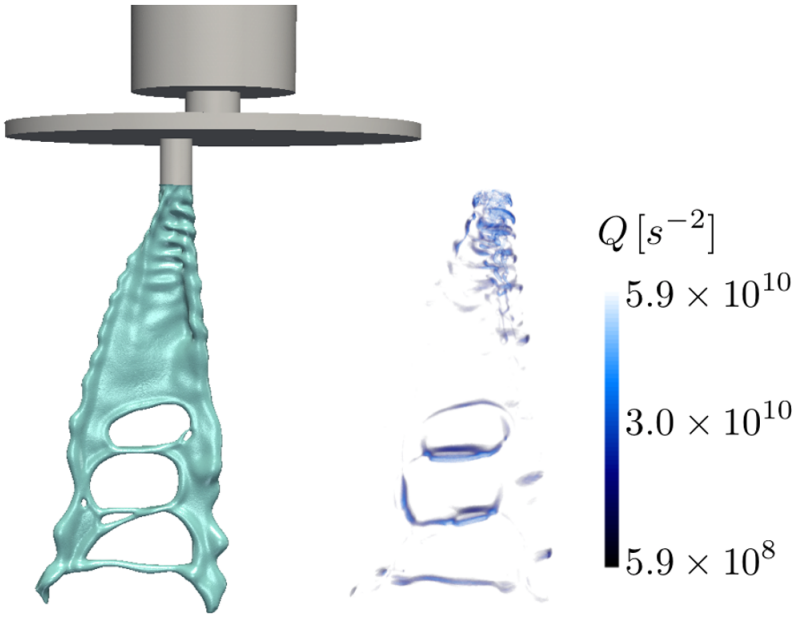
The computational cost and performances of the two conducted simulations on adaptive tetrahedral grids are shown in Table 4.1. The reduced computational time RCT is computed following Eq. (3.8) for each of the major parts of the complete procedure: the interface-capturing technique (ACLS), the remeshing part (AMR), and the linear solver (Poisson). The different values of RCT presented in this table relate to each other as: $\mathrm{RCT} = \mathrm{RCT}_{\mathrm{ACLS}} + \mathrm{RCT}_{\mathrm{AMR}} + \mathrm{RCT}_{\mathrm{Poisson}}$. The CPU time is defined, using the notations of Eq. (3.8), as:

$$\mathrm{CPU\,time} = \mathrm{WCT} \times N_{\mathrm{cores}} = \mathrm{RCT} \times N_{\mathrm{iter}} \times N_{\mathrm{cv}} \tag{4.3}$$

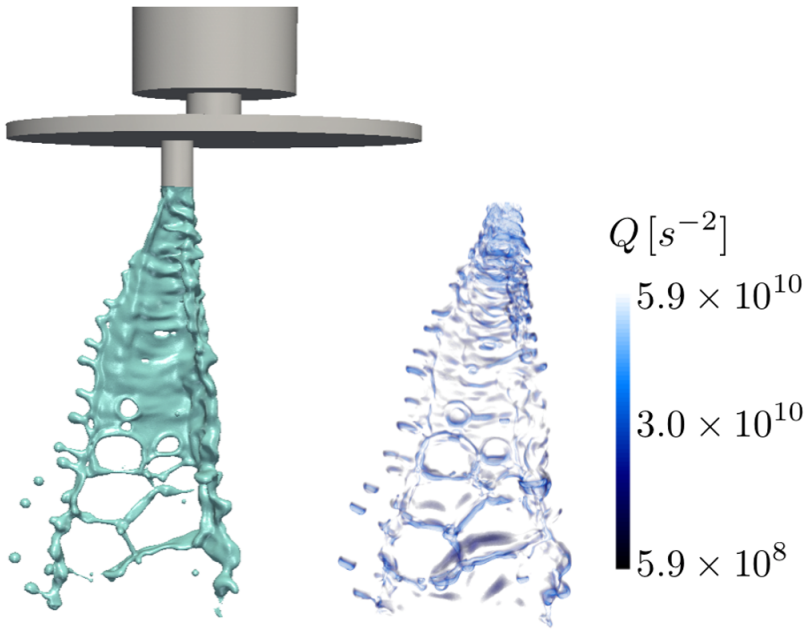
Good scalability of the unstructured ACLS/AMR procedure is displayed. Furthermore, the size of the meshes is moderate: in comparison, the same run at $d_{\mathrm{inj}}/\Delta x_{\min} = 72$ on a static grid requires 8192 cores and $1.62\mathrm{B}$ cells. Hence, the present adaptive method allows huge computational savings for simulations on tetrahedral grids.

Table 4.1: Computational costs and performances of the low-pressure compound injector simulations on adaptive grids. The simulations have been conducted on 2nd Gen AMD EPYC processors.

$d_{\mathrm{inj}}/\Delta x_{\min}$	N_{cores}	N_{cells}	RCT [μs]	$\mathrm{RCT}_{\mathrm{ACLS}}$ [μs]	$\mathrm{RCT}_{\mathrm{AMR}}$ [μs]	$\mathrm{RCT}_{\mathrm{Poisson}}$ [μs]	CPU time [h]/Phys. time [ms]
36	600	48M	287.6	99.7	137.2	50.7	52,174
72	1400	141M	549.8	142.0	208.6	199.2	915,531

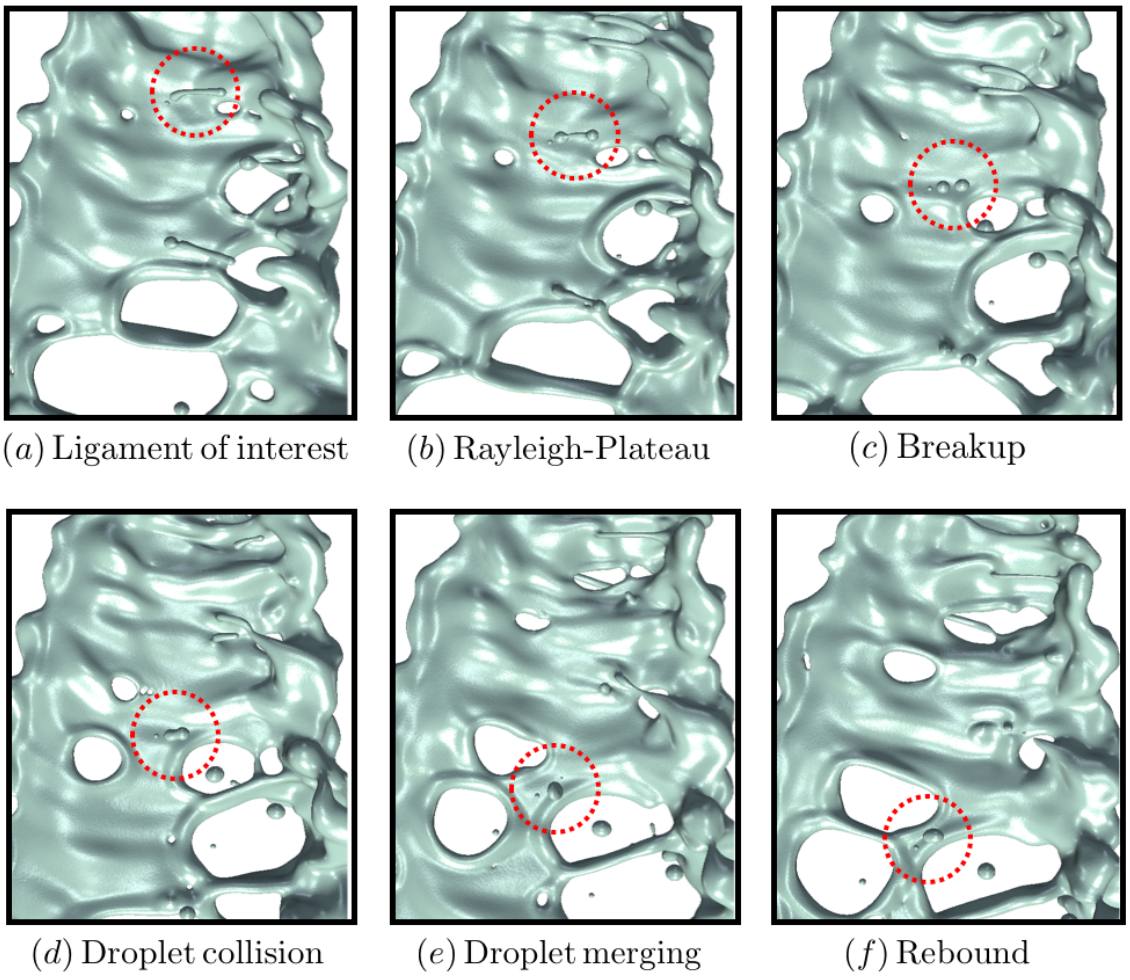


$(a)\, d_{\mathrm{inj}}/\Delta x_{\mathrm{min}} = 36$



$(b)\, d_{\mathrm{inj}}/\Delta x_{\mathrm{min}} = 72$

Figure 4.5: Interface contours and Q-criterion of the water jet atomizing from a low-pressure compound nozzle in quiescent air at the two investigated resolutions: (a) $\Delta x_{\mathrm{min}} = 5\mu\mathrm{m}$ and (b) $\Delta x_{\mathrm{min}} = 2.5\mu\mathrm{m}$.



(a) Ligament of interest (b) Rayleigh-Plateau (c) Breakup

(d) Droplet collision (e) Droplet merging (f) Rebound

Figure 4.6: Closeup on some capillary effects during the atomization process of a water jet in quiescent air from a low-pressure compound nozzle. $d_{\text{inj}} = 180\mu\text{m}$. $\text{Re} = 3653$. $\text{We}_{\text{l}} = 1061$. $\Delta x_{\min} = 2.5\mu\text{m}$.

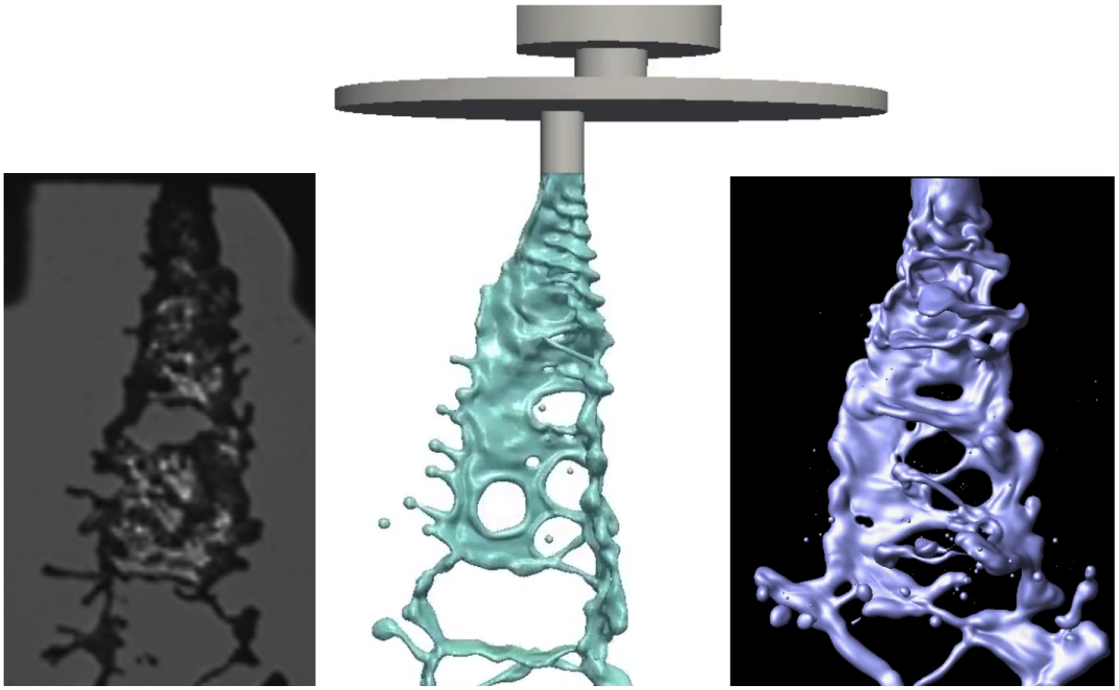


Figure 4.7: Qualitative comparison of the jet topology close to the nozzle between (a) the experiment, (b) the present ACLS/LES study using YALES2 ($\Delta x = 2.5\mu$m at the interface), and (c) the CLSVOF/DNS performed with the Archer code ($\Delta x = 1.44\mu$m). This figure has been built upon the one of [40].

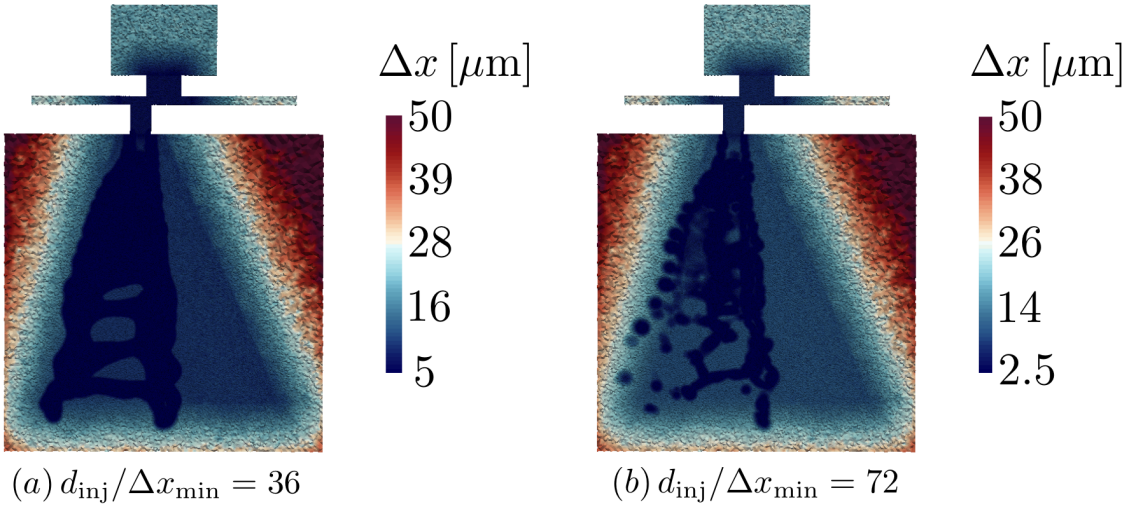
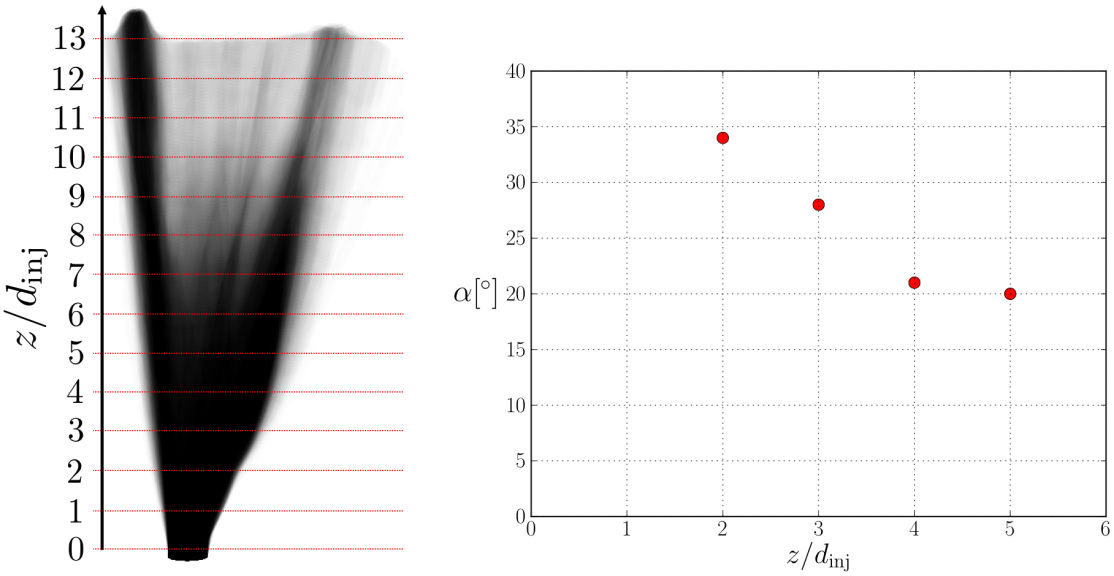


Figure 4.8: Mid-plane colored by cell size. $d_{\text{inj}} = 180\mu$m. (a) $\Delta x_{\text{min}} = 5\mu$m; (b) $\Delta x_{\text{min}} = 2.5\mu$m.



(a) Mean projected spray (b) Spray angle

Figure 4.9: Estimation of the spray angle α at different distances z/d_{inj} from the nozzle exit. $d_{\mathrm{inj}}/\Delta x_{\mathrm{min}} = 72$.

4.2 LES of a high-pressure kerosene jet in crossflow

In order to test the capabilities of the algorithm for resolving atomization in aeronautical injection systems, a non-reactive liquid kerosene jet in crossflow (JICF) injection has been simulated. Liquid is injected through a nozzle into a plenum where a stream of air flows in the perpendicular direction to the liquid (the crossflow). Then, the jet bends towards the air direction and atomizes due to the aerodynamic interaction. The results presented in this section are issued from a joint study between the present work and the one of [180]: the simulations are here exploited as a validation of all algorithms presented earlier in this manuscript, and to assess the computational performances of the various parts of the upgraded YALES2 two-phase flow solver. A more in-depth physical analysis of the case can be found in [180].

The chosen configuration is the experimental test-bench by Becker and Hassa [41], which consists of a kerosene JICF injected into a high-pressure environment representative of the ambient conditions found within aeronautical gas turbines. The physics of the JICF is governed by the two following parameters: the kinetic energy ratio between the liquid and gaseous phases q and the gaseous Weber number We_g. They are defined as follows:

$$\mathrm{q} = \frac{\rho_l u_l^2}{\rho_g u_g^2} \quad \text{and} \quad \mathrm{We}_g = \frac{\rho_g u_g^2 d_{\mathrm{inj}}}{\sigma} \tag{4.4}$$

where ρ is the density, u is the bulk velocity of the flow, d_{inj} is the liquid nozzle diameter ($0.45\mathrm{mm}$ in Fig. 4.10) and σ is the surface tension coefficient. The subindexes l and g denote liquid and gaseous phases respectively.

4.2.1 Numerical setup

The numerical setup is the one of Guillamon et al. (2021) [181]. Fig. 4.10 (a) shows the numerical setup replicating the experiment from [41]. It consists of a plenum of dimensions $(L_x, L_y, L_z) =$

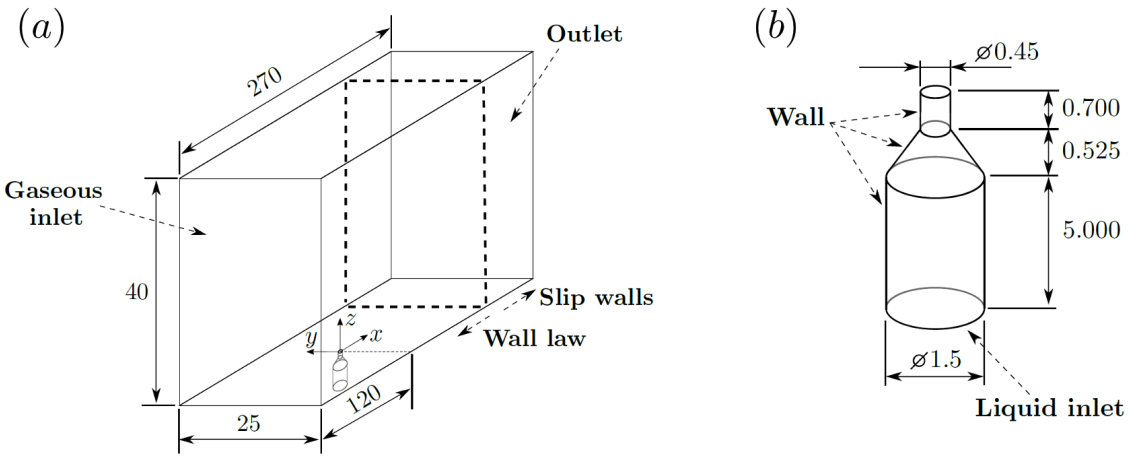


Figure 4.10: Numerical domain and boundary conditions of the experimental test bench of Becker and Hassa [41]: (a) complete domain, (b) detailed view of the injection nozzle. All dimensions are in mm.

$(270\text{mm}, 40\text{mm}, 25\text{mm})$ and a tapered nozzle for liquid injection. A close-up view of the nozzle is shown in Fig. 4.10 (b). The injected liquid is JET-A kerosene with density $\rho_l = 795\text{kg.m}^{-3}$, dynamic viscosity $\mu_l = 1.5 \times 10^{-3}\text{kg.m}^{-1}\text{.s}^{-1}$ and surface tension $\sigma = 22 \times 10^{-3}\text{kg.s}^{-2}$. The gaseous crossflow is pressurized air at 6bar with density $\rho_g = 7.21\text{kg.m}^{-3}$ and dynamic viscosity $\mu_g = 1.82 \times 10^{-5}\text{kg.m}^{-1}\text{.s}^{-1}$. The density and dynamic viscosity ratios are respectively $\rho_l/\rho_g = 110$ and $\mu_l/\mu_g = 83$. The Ohnesorge number is $\text{Oh} = \mu_l/\sqrt{\rho_l \sigma d_{\text{inj}}}$ = 0.017. Two operating points studied experimentally by [41] are simulated. They are given by the dimensionless numbers q and We_g shown in Table 4.2. Two interface resolutions are considered for both operating points: $\Delta x_{\min} = 20\mu\text{m}$ and $10\mu\text{m}$, that can be expressed relative to the injection diameter as $d_{\text{inj}}/\Delta x_{\min}$ = 22.5 and 45 respectively. Therefore, a total of four JICF cases are computed. These simulations are performed with LES where a dynamic Smagorinsky closure is used for modeling the unresolved turbulence scales [70, 71].

Table 4.2: JICF operating points

Operating point	$u_l[\text{m.s}^{-1}]$	$u_g[\text{m.s}^{-1}]$	Re_g	q	We_g	$d_{\text{inj}}/\Delta x_{\min}$
1	17.5	75	0.92×10^6	6	830	22.5; 45
2	23.33	100	1.22×10^6	6	1470	22.5; 45

For kerosene injection, a Poiseuille profile is prescribed at the liquid inlet. The mean velocity of this profile is equal to the bulk velocity u_l given in Table 4.2. Regarding the gaseous phase, the experiments from [41] report a gaseous boundary layer thickness of 5mm right upstream the injector. As the numerical domain is smaller than the experimental test bench, a velocity profile with near-wall and outer regions is specified at the gaseous inlet. The near-wall region is modeled considering that the gaseous phase evolves in the streamwise direction as a turbulent boundary layer along a flat plate. Its thickness is calculated following a 1/7th power law. The outer part of the velocity profile has a constant value chosen so that the injected mass flow rate matches the one reported in the experiments. Additionally, synthetic turbulent fluctuations are added to the mean velocity profiles since the Reynolds numbers at the gaseous inlet Re_g shown in Table 4.2 indicate

turbulent flow.

The initial mesh is the same for all simulations: it consists of 66M tetrahedral elements ($N_{\text{cells}} = 66\text{M}$). The number of cores N_{cores} allocated to each simulation ensures a constant ratio $N_{\text{cells}}/N_{\text{cores}}$ between 100,000 and 150,000. For all simulations, the refined region is of half-width $10\Delta x_{\min}$ ($N_p = 10$, see Fig. 2.8), and AMR is triggered each time the interface is 4 cells away from the edge of this region. Each case will yield different meshes as function of the operating condition and mesh resolution, see Table 4.4. Therefore, the simulations will have different costs, see Table 4.5. Physical running times simulated depend on the computational resources available.

4.2.2 Jet topology and breakup

The effect of mesh resolution (see Table 4.4) is shown in the instantaneous snapshots of Fig. 4.11 for the operating point $\text{We}_g = 1470$. The liquid-gas interface and the mesh at the plane $y = 0$ are displayed. As for the atomizing spray issued from a low-pressure compound nozzle (Section 4.1), the interface cell size has a direct effect onto breakup and atomization. Surface breakup mechanisms [41], characterized by corrugations in the dense core surface, are observed for both simulations but much less pronounced for the coarse case. Column instabilities [41] are clearly visible in the fine case, but barely apparent in the coarse simulation. These differences between the two interface resolutions overall lead to much more droplets produced by the primary atomization process in the fine simulation (see Table 4.4), as it can be seen from the different liquid ligaments and structures that result from the breakup of the liquid dense core. Moreover, the smallest droplets directly issued from the dense core (surface breakup) are more susceptible to remain tracked using the fine adaptive grid (less mass losses). Subsequent secondary atomization may also be affected by the resolution, as the finest mesh can capture smaller droplets than the coarse grid (see Table 4.4).

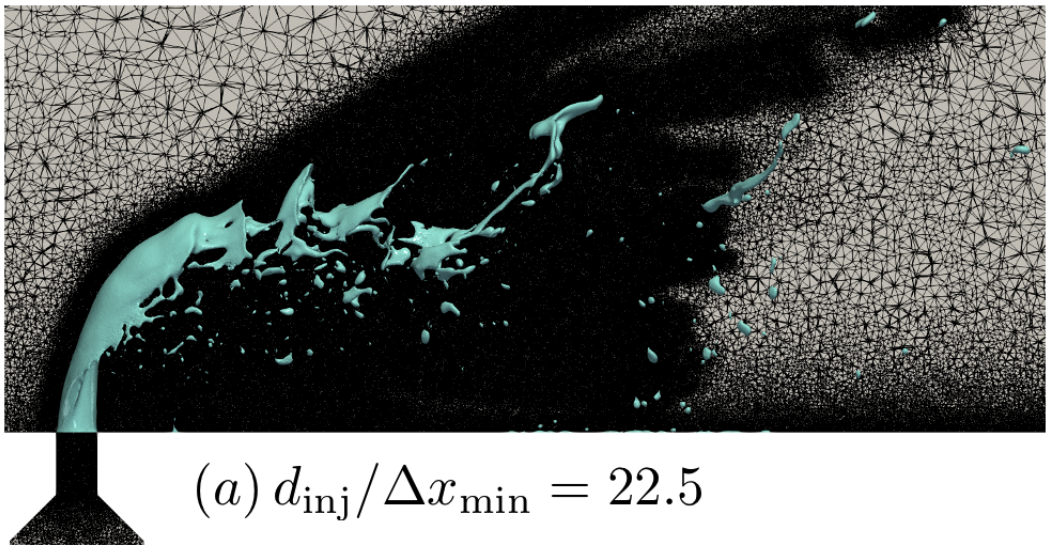
Regarding the operating point, Fig. 4.12 shows instantaneous snapshots of the liquid-gas interface for the finest case ($d_{\text{inj}}/\Delta x_{\min} = 45$), for both values of We_g. The same time instant than in Fig. 4.11 is displayed. By comparing the operating points, it is seen that both cases present column breakup features, but more pronounced for $\text{We}_g = 830$: this simulation indeed shows that the ligaments formed during primary atomization stretch and form lobes more pronounced (see Figs. 4.12 (a) and (b)) than for $\text{We}_g = 1470$. The operating point $\text{We}_g = 1470$ features a higher gas velocity, inducing more lateral instabilities (i.e. more surface breakup) due to the strong shear force exerted by the incoming air [180]: the dense core is thus thiner as seen in Fig. 4.12 (c), compared to Fig. 4.12 (a). It is clear by comparing Figs. 4.12 (b) and (d) that the operating point with the highest gaseous Weber number produces more droplets: this is confirmed by Table 4.4.

4.2.3 Vertical trajectory and experimental validation

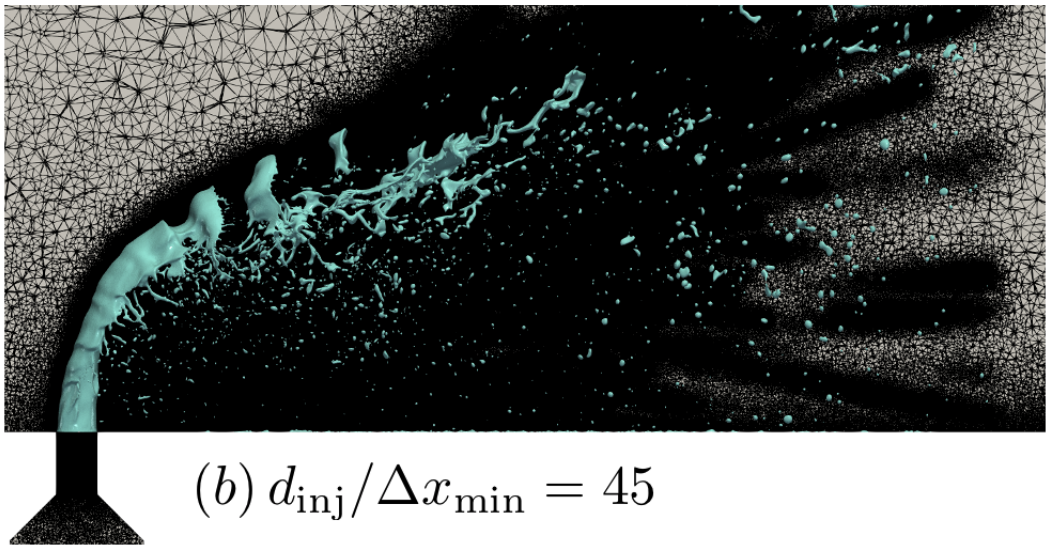
Numerical simulations are validated with the experimental correlation obtained by [41] for the vertical jet penetration. It has been obtained by testing experimentally several operating conditions, and has a standard deviation of value 0.81. The correlation corresponds to the trajectory of the jet's windward side, and is given by:

$$\frac{z}{d_{\text{inj}}} = 1.57\text{q}^{0.36} \ln\left(1 + 3.81\frac{x}{d_{\text{inj}}}\right) \tag{4.5}$$

This trajectory depends solely on the q factor, as the authors state that the influence of the Weber number on the trajectory is negligible. Other experimental studies, such as the one by Ragucci [42],

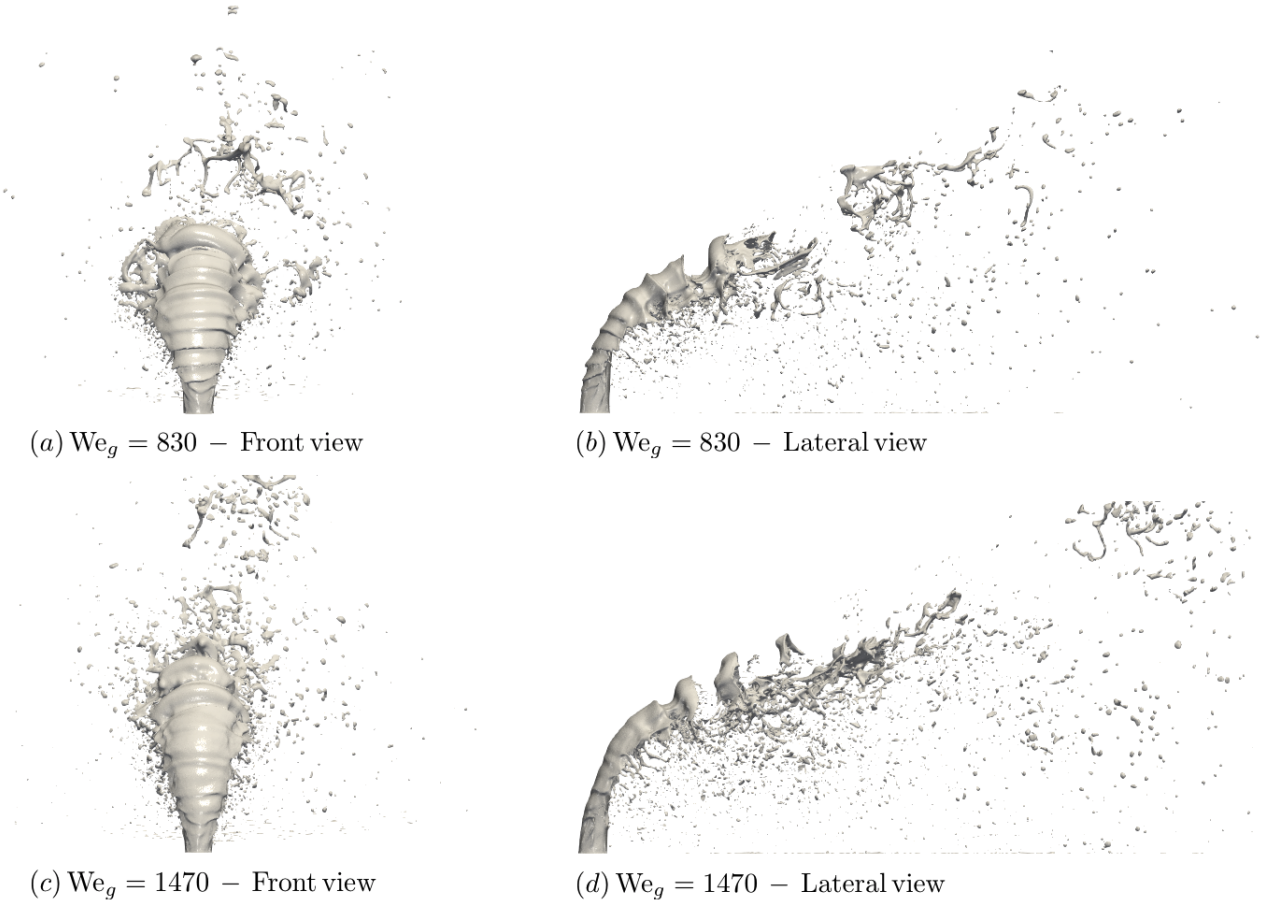


$(a)\, d_{\mathrm{inj}}/\Delta x_{\mathrm{min}} = 22.5$



$(b)\, d_{\mathrm{inj}}/\Delta x_{\mathrm{min}} = 45$

Figure 4.11: Lateral view of meshes and interface contours near the injector at time instant $t = 0.3$ms. $\mathrm{q} = 6$. $\mathrm{We}_g = 1470$.



(a) $\mathrm{We}_g = 830$ – Front view

(b) $\mathrm{We}_g = 830$ – Lateral view

(c) $\mathrm{We}_g = 1470$ – Front view

(d) $\mathrm{We}_g = 1470$ – Lateral view

Figure 4.12: Instantaneous front (left) and lateral (right) views at time instant $t = 0.3$ms for the two operating conditions. $d_{\mathrm{inj}}/\Delta x_{\mathrm{min}} = 45$. q $= 6$. $\mathrm{We}_g = 830$ (top), $\mathrm{We}_g = 1470$ (bottom).

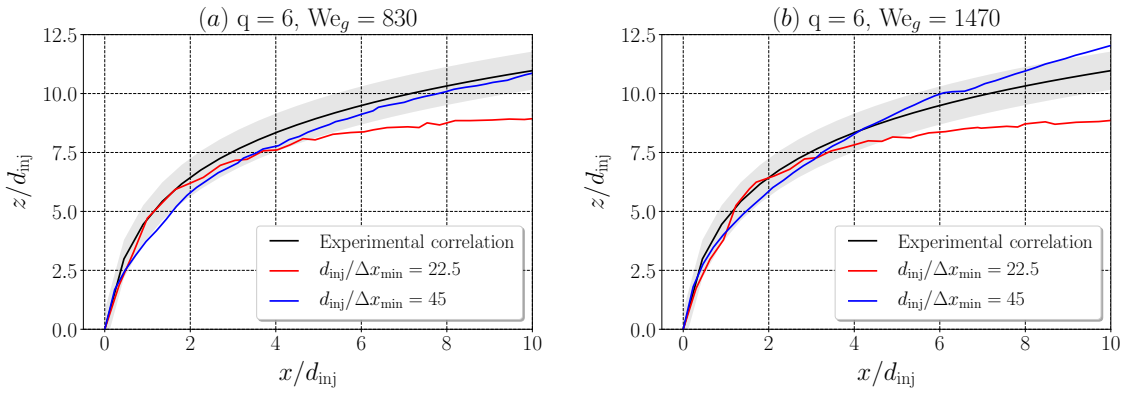


Figure 4.13: Numerical mean jet trajectories for (a) $\text{We}_g = 830$ and (b) $\text{We}_g = 1470$ compared with the experimental correlation from Becker [41]. The grey area represents the standard deviation of the experimental trajectory correlation provided by [41].

provide experimental correlations with a dependence on We. While the role of We on the trajectory is still an open question, the results of our simulations are only compared to the ones of [41] (Eq. (4.5)).

The resulting mean trajectories are shown in Fig. 4.13. The experimental correlation is given by the black solid line, and the shadowed area denotes its confidence region. The red and blue lines are the numerical mean trajectories for the coarse and fine mesh resolutions, $d_{\text{inj}}/\Delta x_{\text{min}} = 22.5$ and $d_{\text{inj}}/\Delta x_{\text{min}} = 45$, respectively. These trajectories have been obtained by calculating the location z/d_{inj} of the maximum value of the gradient in the z direction of the time-averaged level-set function, $\max\left(\nabla_z \overline{\psi\left(\mathbf{x}\right)}\right)$, for each value of the abscissa x/d_{inj}. This method has been chosen as it is based on the experimental methodology used by [41] for obtaining the experimental correlations. Fig. 4.13 (a) shows the $\text{We}_{\text{g}} = 830$ operating point from Table 4.2. The trajectory from the coarse mesh is accurately estimated in the near-nozzle region and gets underestimated further downstream for $x/d_{\text{inj}} > 5$. The fine resolution shows a better approach to experiments, always comprised within the confidence interval of the experimental correlation. Regarding the operating point We_g = 1470, shown in Fig. 4.13 (b), the same trends are observed: the coarse mesh underestimates the trajectory downstream, while the fine mesh shows good experimental comparison within all the displayed range. The numerical trajectories show in some regions a non-smooth behavior since full convergence has not been achieved in all cases due to the high cost of the simulations (see Table 4.4). This occurs specially from $x/d_{\text{inj}} > 2.5$ in Fig. 4.13 (a) for $d_{\text{inj}}/\Delta x_{\text{min}} = 22.5$ and in Fig. 4.13 (b): closer to the injector, the dense core is present and the continuous presence of liquid yields smooth trajectories, while further away atomization produces ligaments and droplets that hinder the convergence of mean trajectories.

The accuracy of the mean numerical trajectories can be quantitatively assessed by defining a L_2 error as in Eq. (4.6):

$$L_2 = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{z}{d_{\text{inj}}}\bigg|_{\text{exp},i} - \frac{z}{d_{\text{inj}}}\bigg|_{\text{num},i}\right)^2} \tag{4.6}$$

where N is the total number of sample points along the trajectory coordinate x/d_{inj}. The values for each case are shown in Table 4.3. For both operating points, the coarse resolution show similar and larger errors than for the fine case due to the underestimation of the experimental trajectory

further downstream. The fine resolutions show lower errors, with agrees with the observations of Fig. 4.13. The simulation for the low We_g and fine resolution yields the lowest error of all.

Table 4.3: Error levels of the L_2 norm (Eq. (4.6)) for jet in crossflow simulations

We_g	$d_{\mathrm{inj}}/\Delta x_{\mathrm{min}}$	L_2
830	22.5	1.147
830	45	0.501
1470	22.5	1.182
1470	45	0.544

4.2.4 Computational performances

Table 4.4 shows the number of cores, tetrahedral elements and droplets generated at $t = 0.3\mathrm{ms}$ for all simulations. By comparing the mesh resolutions for each operating point, it is observed that six times more droplets are generated when the minimum cell size is reduced twice. Thus, as more droplets are resolved, more cores are needed to keep the ratio $N_{\mathrm{cells}}/N_{\mathrm{cores}}$ constant for optimal performances: when the resolution increases twice, the number of elements rises near fivefold. When the operating points are compared at identical mesh resolutions, around 50% more droplets are present in the highest gaseous-Weber case. This is due to a higher crossflow velocity, which increases the number of droplets generated by surface breakup.

The computational performances and costs are shown in Table 4.5. The Reduced Computational Times (RCT) are calculated according to Eq. (3.8) for each of the major parts of the complete procedure: the interface-capturing technique (ACLS), the remeshing part (AMR), and the linear solver (Poisson). They relate to each other as: $\mathrm{RCT} = \mathrm{RCT}_{\mathrm{ACLS}} + \mathrm{RCT}_{\mathrm{AMR}} + \mathrm{RCT}_{\mathrm{Poisson}}$. The CPU time is defined following Eq. (4.3). In all cases, the time spent in the ACLS procedure is almost constant and moderate despite the large liquid-gas surface to be resolved, demonstrating the high scalability of the interface-capturing algorithm. Most of the total computational time is spent in the AMR routine: indeed, the jet velocity is high and hence the AMR is automatically triggered much more frequently than in the low-pressure jet simulations of Section 4.1. In general, jet in crossflow simulations are more expensive than the low-pressure jet ones (see Table 4.1 for comparison): while this latter jet is governed by capillary forces and aerodynamic effects are negligible, the jet in crossflow atomization is driven by aerodynamics and inertia (the limiting timestep of the simulations is always due to the CFL number).

It can be concluded that most of the computational resources allocated to the jet in crossflow cases are used for solving the AMR routine, making these computations highly expensive for cases where very small droplets are generated and a higher resolution of the atomization is required. A promising solution to this problem is coupling the presented methodology to a Lagrangian solver that could identify spherical droplets and convert them to Lagrangian Point Particles (LPP). This Eulerian-Lagrangian two-way coupling would allow spherical droplets to move following a ballistic motion and break up according to secondary atomization models. In this way, droplets would not be resolved with the Eulerian mesh by the ACLS/AMR methodology, saving huge computational resources.

Table 4.4: Number of computational cores, mesh cells and droplets generated for jet in crossflow simulations. All values correspond to time instant $t = 0.3$ms.

We_g	$d_{\mathrm{inj}}/\Delta x_{\mathrm{min}}$	N_{cores}	N_{cells}	N_{drops}
830	22.5	768	100M	394
830	45	3840	430M	2221
1470	22.5	1024	138M	596
1470	45	5760	830M	4153

Table 4.5: Computational performances for jet in crossflow simulations at time instant $t = 0.3$ms.

We_g	$d_{\mathrm{inj}}/\Delta x_{\mathrm{min}}$	RCT [μs]	$\mathrm{RCT}_{\mathrm{ACLS}}$ [μs]	$\mathrm{RCT}_{\mathrm{AMR}}$ [μs]	$\mathrm{RCT}_{\mathrm{Poisson}}$ [μs]	CPU time [h]/Phys. time [ms]
830	22.5	494.1	88.8	362.3	43.0	17,100
830	45	1549.7	192.0	1182.6	175.1	204,200
1470	22.5	1061.5	145.3	838.0	78.2	28,700
1470	45	1760.7	164.1	1476.7	119.9	413,000

Part II

Multiscale modeling on adaptive unstructured grids

CHAPTER 5

Lagrangian Particle Tracking for two-phase liquid-gas flows

Contents

5.1 Governing equations of the Euler-Lagrange framework 117
5.1.1 Equations of motion of Lagrangian point particles 117
5.1.2 One-way coupling approximation 120
5.1.3 Influence of the particles on the flow: two-way coupling 120
5.2 Conversion between Eulerian and Lagrangian frameworks 122
5.2.1 Identification of isolated Eulerian liquid structures 122
5.2.2 Conversion criteria between Eulerian and Lagrangian frameworks 122
5.2.3 Coupling with Adaptive Mesh Refinement 125
5.2.4 Summary of the conversion procedure 125

The first part of this manuscript raised the need to alleviate several computational constraints in atomization simulations, especially the jet in crossflow configuration. First, the large number of droplets produced by the atomization process increases significantly the size of the grid, even with Adaptive Mesh Refinement (AMR). This yields to prohibitive computational costs, especially when it comes to validation against experiments far from the injection location. Then, in such simulations driven by inertia, high-velocity drops may force the triggering of AMR very often, leading again to an increase in computational cost. Eventually, the smallest drops in the domain are poorly resolved using classical interface-capturing techniques, even with AMR, and mass losses (even if reasonable) occur using the ACLS method. This second part is an attempt to address these issues, still using adaptive unstructured grids. The goal is to couple the adaptive Eulerian methodology, presented in the first part of this thesis, to a Lagrangian modeling technique for the small spherical drops close to resolution limit. The small (in terms of effective resolution) and spherical Eulerian droplets need to be converted into point particles dynamically, throughout the simulations and hence robust conversion criteria are needed for such hybrid Euler-Lagrange technique. The use of AMR paves the way to a robust and efficient multi-scale strategy, allowing to target the optimal conditions for both frameworks: well-refined Eulerian drops, and strong coarsening of the mesh around the newly-introduced particles to meet the Lagrangian assumptions and automatically lighten the grid.

5.1 Governing equations of the Euler-Lagrange framework

5.1.1 Equations of motion of Lagrangian point particles

The first theoretical approach for describing the motion of a rigid spherical particle has been conducted by Stokes (1851) [182], in which a point particle is immersed in a viscous fluid with a

low relative velocity. By writing Newton's second law of motion for the particle in the presence of dominant viscous forces over inertia (i.e. low Reynolds number $\mathrm{Re} \ll 1$) [183]:

$$m_p \frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} = 3\pi d_p \nu \rho (\tilde{\mathbf{u}}(\mathbf{x}_p) - \mathbf{u}_p) \tag{5.1}$$

with m_p, d_p, and $\mathbf{u}_p$ the particle mass, diameter, and velocity, respectively. ν and ρ are the fluid kinematic viscosity and density, respectively. $\tilde{\mathbf{u}}(\mathbf{x}_p)$ is the undisturbed fluid velocity at the particle position (i.e. at the particle mass center). $\mathbf{F}_{D,\mathrm{St}} = 3\pi d_p \nu \rho (\tilde{\mathbf{u}}(\mathbf{x}_p) - \mathbf{u}_p)$ is the well-known Stokes' drag force. One can thus write the Stokes dimensionless number St, which represents the ratio between the particle characteristic time τ_p and the shortest turbulent flow characteristic time τ_d (dissipative time scale). These time scales write, respectively [183]:

$$\tau_p = \frac{\rho_p}{\rho} \frac{d_p^2}{18\nu} \quad \text{and} \quad \tau_d = \frac{l_d^2}{\nu} \tag{5.2}$$

with ρ_p the particle density, and l_d the turbulent dissipative length scale (Kolmogorov scale). The Stokes number then yields [183]:

$$\mathrm{St} = \frac{\tau_p}{\tau_d} = \frac{1}{18} \frac{\rho_p}{\rho} \left(\frac{d_p}{l_d} \right)^2 \tag{5.3}$$

The asymptotic behaviors of the particle provided by the Stokes number are [183]:

- $\mathrm{St} \ll 1$: the particle time scale is much shorter than the turbulent flow dissipative time scale ($\tau_p \ll \tau_d$). The particle follows perfectly the flow streamlines and is thus a tracer: the particle trajectory is driven by the flow dynamics;
- $\mathrm{St} \gg 1$: the particle time scale is much longer than the turbulent flow dissipative time scale ($\tau_p \gg \tau_d$). The particle trajectory does not follow the flow streamlines.

Later, based on the work of Maxey and Riley (1983) [184], and Gatignol (1983) [185], a rigid spherical Lagrangian Point Particle (LPP) p with a constant mass $m_p = \rho_p V_p$, position $\mathbf{x}_p$ and velocity $\mathbf{u}_p$, moving in a surrounding gas flow of constant density ρ_g, constant dynamic viscosity μ_g, and velocity $\mathbf{u}$, follows the general equation:

$$m_p \frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} = \mathbf{F}_I + \mathbf{F}_G + \mathbf{F}_D + \mathbf{F}_A + \mathbf{F}_H \tag{5.4}$$

which is derived from Newton's second law of motion. In this equation, the various forces $\mathbf{F}$ are:

- $\mathbf{F}_I = \rho_g V_p \left(\frac{\partial \tilde{\mathbf{u}}(\mathbf{x}_p)}{\partial t} + \boldsymbol{\nabla} \tilde{\mathbf{u}}(\mathbf{x}_p) \cdot \tilde{\mathbf{u}}(\mathbf{x}_p) \right)$ is the inertial force, where $\tilde{\mathbf{u}}(\mathbf{x}_p)$ is the undisturbed gas velocity at the particle position;
- $\mathbf{F}_G = (\rho_p - \rho_g) V_p \mathbf{g}$ is the gravitational force;
- $\mathbf{F}_D$ is the drag force;
- $\mathbf{F}_A$ is the added-mass force, which accounts for the relative acceleration of the surrounding gas;
- $\mathbf{F}_H$ is the history force, arising from the time necessary to develop the boundary layer around the droplet [186, 48].

In this work, particles represent droplets, and thus large liquid-gas density ratios are involved: $\rho_p = \rho_l \gg \rho_g$. As a consequence, the inertial force, the added-mass force, and the history force can be neglected [187, 188]. These simplifications eventually lead to the following set of equations governing the trajectory of the mass center of a particle p:

$$\boxed{\begin{aligned} \frac{\mathrm{d}\mathbf{x}_p}{\mathrm{d}t} &= \mathbf{u}_p \\ m_p \frac{\mathrm{d}\mathbf{u}_p}{\mathrm{d}t} &= \mathbf{F}_G + \mathbf{F}_D \end{aligned}} \tag{5.5, 5.6}$$

With the assumption $\rho_p \gg \rho_g$, the buoyancy effect may be neglected and the gravitational force $\mathbf{F}_G$ now writes:

$$\boxed{\mathbf{F}_G = \rho_p V_p \mathbf{g} = m_p \mathbf{g}} \tag{5.7}$$

The drag force $\mathbf{F}_D$ is:

$$\mathbf{F}_D = m_p \frac{\tilde{\mathbf{u}}(\mathbf{x}_p) - \mathbf{u}_p}{\tau_p} \tag{5.8}$$

with τ_p the response time scale of the particle, which depends on the drag coefficient C_D and particle Reynolds number Re_p:

$$\tau_p = \frac{4}{3 C_D \mathrm{Re}_p} \frac{\rho_p}{\rho_g} \frac{d_p^2}{\nu_g} \tag{5.9}$$

Using Eq. (5.9), Eq. (5.8) writes:

$$\boxed{\mathbf{F}_D = \frac{\pi}{8} d_p \mu_g C_D \mathrm{Re}_p (\tilde{\mathbf{u}}(\mathbf{x}_p) - \mathbf{u}_p)} \tag{5.10}$$

Re_p, the Reynolds number based on the particle diameter, is defined at the Lagrangian point-particle as:

$$\mathrm{Re}_p = \frac{\rho_g |\tilde{\mathbf{u}}(\mathbf{x}_p) - \mathbf{u}_p| d_p}{\mu_g} \tag{5.11}$$

The drag coefficient C_D strongly varies with Re_p, and thus necessitates a closure. The correlation of Schiller and Naumann [189], which extends Stokes' drag law, is used in this work. Stokes' law is applied when $\mathrm{Re}_p < 1$, the empirical correlation of Schiller and Naumann if $1 < \mathrm{Re}_p < 1000$, and a constant value due to the sufficiently developed turbulence behind the particle for large particle Reynolds numbers $\mathrm{Re}_p > 1000$:

$$C_D = \begin{cases} \frac{24}{\mathrm{Re}_p}, \text{ if } \mathrm{Re}_p < 1 \\ \frac{24}{\mathrm{Re}_p} \left(1 + 0.15 \mathrm{Re}_p^{0.687}\right), \text{ if } 1 < \mathrm{Re}_p < 1000 \\ 0.44, \text{ if } \mathrm{Re}_p > 1000 \end{cases} \tag{5.12}$$

A particle Weber number can also be defined:

$$\mathrm{We}_p = \frac{\rho_g |\tilde{\mathbf{u}}(\mathbf{x}_p) - \mathbf{u}_p|^2 d_p}{\sigma} \tag{5.13}$$

It should be noted that the liquid-gas flows studied in this work are considered in the dilute regime. As a consequence, particle-particle interactions are neglected: particle collisions and scaling of drag force with volume fraction will not be considered.

As briefly seen earlier in this section, the resolved gaseous velocity field $\mathbf{u}$ can be decomposed into an undisturbed velocity field $\tilde{\mathbf{u}}$, and a velocity disturbance $\mathbf{u}'$ induced by the presence of the particle considered. This idea has been suggested first by Maxey et al. (1997) [190] and writes:

$$\mathbf{u}(\mathbf{x}_p) = \tilde{\mathbf{u}}(\mathbf{x}_p) + \mathbf{u}'(\mathbf{x}_p) \tag{5.14}$$

The velocity disturbance $\mathbf{u}'$ accounts for the mechanical action of the Lagrangian particle on the resolved gaseous flow field.

5.1.2 One-way coupling approximation

In the one-way coupling approximation, the influence of the particles on the gaseous flow field is neglected: $\mathbf{u} = \tilde{\mathbf{u}}$ and $\mathbf{u}' = \mathbf{0}$. As a consequence, the momentum contribution of the Lagrangian particles on the Eulerian flow is neglected. The validity of the one-way approximation is related to the mass fraction of the particles [191, 192, 193, 27]: it is indeed valid when there are only a few particles in the gaseous flow field. For instance, according to Crowe et al. (1998) [191], one can assess the importance of momentum coupling for canonical flows by estimating the ratio $\Pi_{\mathcal{M}}$ between the drag force F_D due to the particles in a certain volume $V = L^3$ and the momentum flux $\phi_{\mathcal{M}}$ of the carrier phase through the volume. For a Stokes flow, after many developments, the momentum coupling parameter yields:

$$\Pi_{\mathcal{M}} = \frac{\mathrm{F}_D}{\phi_{\mathcal{M}}} = \frac{C}{1 + \mathrm{St}_{\mathcal{M}}} \tag{5.15}$$

with $\mathrm{St}_{\mathcal{M}} = (\tau_p \mathrm{u})/L$ the Stokes number for momentum transfer, and C the mass concentration of particles, defined as:

$$C = \frac{\alpha_p \rho_p}{(1 - \alpha_p)\rho} \tag{5.16}$$

where $\alpha_p = V_p/V$ is the volume fraction of the particles. Hence, the momentum coupling effects are less important for small mass concentrations of particles.

The one-way coupling approximation is used in all simulations presented in this work.

5.1.3 Influence of the particles on the flow: two-way coupling

In order to account for the influence of the particles on the flow, a two-way coupling momentum source term $\mathbf{\Phi}_{p\to f}$ needs to be added in the momentum equation, which now writes:

$$\boxed{\frac{\partial(\rho\mathbf{u})}{\partial t} + \boldsymbol{\nabla}\cdot(\rho\mathbf{u}\otimes\mathbf{u}) = -\boldsymbol{\nabla}p + \boldsymbol{\nabla}\cdot\boldsymbol{\tau} + \rho\mathbf{g} + \mathbf{\Phi}_{p\to f}} \tag{5.17}$$

The momentum source term is expressed as:

$$\mathbf{\Phi}_{p\to f} = \sum_{p=1}^{N_p} -\mathbf{F}_{D,p}\delta(\mathbf{x} - \mathbf{x}_p) \tag{5.18}$$

Using the Particle-Source-In-Cell (PSIC) model of Crowe (1977) [194] to approximate the Dirac-delta function $\delta(\mathbf{x} - \mathbf{x}_p)$, the momentum source term is now written as:

$$\mathbf{\Phi}_{p\to f} = \frac{1}{V}\sum_{p=1}^{N_p} -\mathbf{F}_{D,p} \tag{5.19}$$

where N_p is the number of particles acting on the control volume V. The computation of this backward-coupling term requires interpolation of particle data on the Eulerian grid, to project back the particle drag force to the fluid grid. As mentioned in the latest review of Brandt and Coletti (2022) [195], this is problematic when the particles are not much smaller than the local grid spacing, and when they are so dilute to produce a stiff force field [192]. Indeed, a well-known problem when performing two-way coupled Euler-Lagrange modeling is that the velocity field available on the Eulerian grid is the disturbed velocity field $\mathbf{u}$, not the undisturbed field $\tilde{\mathbf{u}}$. If $\mathbf{u} = \tilde{\mathbf{u}}$ is set to compute the drag force $\mathbf{F}_D$ as in the one-way coupling approximation, an error is introduced. This error relates to the magnitude of the velocity disturbance $\mathbf{u}' = \mathbf{u} - \tilde{\mathbf{u}}$, which is itself related to the particle diameter/cell size ratio $d_p/\Delta x$. This explains why it is necessary to keep the local cell size much larger than the particle size: $d_p \ll \Delta x$. A detailed quantification of these errors using the PSIC model, for various particle Reynolds numbers and $d_p/\Delta x$, have been performed by Evrard et al. (2021) in [196]. For example, in the Stokes regime, the relative error is of order $(6/5) \times (d_p/\Delta x)$.

To reduce these errors for higher particle diameter/cell size ratios (typically $1 < d_p/\Delta x < 10$), Capecelatro and Desjardins (2013) [197] introduced a Volume-Filetered (VF) formulation of the momentum exchange term:

$$\mathbf{\Phi}_{p\to f} = \sum_{p=1}^{N_p} -G\left(|\mathbf{x} - \mathbf{x}_p|\right) \mathbf{F}_{D,p} \tag{5.20}$$

where $G\left(|\mathbf{x} - \mathbf{x}_p|\right)$ is a Gaussian filter kernel, to regularize the backward projection of the drag force to the fluid grid.

More recently, to reduce the errors even more for arbitrary $d_p/\Delta x$ values, Evrard et al. (2020) [198] proposed a corrected VF formulation, using a Wendland filter kernel (whose length scale relates to d_p), by accounting for the particle-induced velocity disturbance $\mathbf{u}'$. This method allows to recover the locally-averaged undisturbed velocity field $< \tilde{\mathbf{u}} > (\mathbf{x}_p)$ by substracting the velocity disturbance contribution (correction term) from the locally-averaged gaseous flow field $< \mathbf{u} > (\mathbf{x}_p)$. The advantage of this approach is that it converges with mesh refinement.

Another approach is to perform a strong coarsening of the grid around the particle, to keep $d_p \ll \Delta x$ so that the errors on the drag force remain as small as possible. In this case filtering is not necessary. This approach is prefered in this work, as the Adaptive Mesh Refinement (AMR) procedure proposed in the first part of the present manuscript allows to perform a strong local coarsening of the grid. Despite the only use of the one-way coupling approximation in this study, the coupling with AMR is implemented in this work and constitutes a start towards future two-way coupled Euler-Lagrange simulations.

To sum-up this Section 5.1, the Lagrangian modeling framework that will be used to represent spherical droplets issued from atomization and advected in a surrounding gas flow in the hybrid Euler-Lagrange simulations has been presented. The matter which now needs to be addressed is: *How to bridge the adaptive unstructured Eulerian framework, presented in the first part of this thesis, and the Lagrangian framework?* In other words, one now needs to address the conversion methodology between the frameworks and its associated criteria to yield a robust and efficient multi-scale strategy using unstructured grids.

5.2 Conversion between Eulerian and Lagrangian frameworks

5.2.1 Identification of isolated Eulerian liquid structures

In order to uniquely detect and identify each separated Eulerian liquid structure in the domain, a parallel algorithm suited for the unstructured grids used in this work, similar to the one of Herrmann (2010) [25], is employed. Using this tagging procedure, each liquid structure in the domain has a unique identificator (called "color" in the present work), noted $\mathcal{C}_d = 1...N_d$ in the following. From the algorithmic point of view, it allows to loop over each liquid droplet and compute its characteristics.

The volume of an isolated Eulerian liquid droplet V_d is integrated as follows:

$$V_d = \int_V \psi \, dV \simeq \sum_{i \in \text{band}} \psi_i V_i \tag{5.21}$$

where ψ is the ACLS hyperbolic tangent function, V is the control volume, and $i \in \text{band}$ is the node considered in the narrow band flagged around the interface of the liquid structure of interest. The droplet equivalent diameter d_d can be computed directly from V_d:

$$\mathrm{d}_d = \left(\frac{6 V_d}{\pi} \right)^{\frac{1}{3}} \tag{5.22}$$

The position $\mathbf{x}_d$ and velocity $\mathbf{u}_d$ of the Eulerian droplet are then calculated according to the following equations, respectively:

$$\mathbf{x}_d = \frac{1}{V_d} \int_V \psi \mathbf{x} \, dV \simeq \frac{\sum_{i \in \text{band}} \psi_i \mathbf{x}_i V_i}{\sum_{i \in \text{band}} \psi_i V_i} \tag{5.23}$$

and

$$\mathbf{u}_d = \frac{1}{V_d} \int_V \psi \mathbf{u} \, dV \simeq \frac{\sum_{i \in \text{band}} \psi_i \mathbf{u}_i V_i}{\sum_{i \in \text{band}} \psi_i V_i} \tag{5.24}$$

In order to evaluate the sphericity of the liquid structure, its aspect ratio γ_d can also be computed:

$$\gamma_d \simeq \frac{\min_{ik \in \text{cp}} |\mathbf{x}_{\Gamma,ik} - \mathbf{x}_d|}{\max_{ik \in \text{cp}} |\mathbf{x}_{\Gamma,ik} - \mathbf{x}_d|} \tag{5.25}$$

with $ik \in \text{cp}$ the pair of nodes ik among the set of crossing pairs cp, $\mathbf{x}_d$ the volume center of the droplet computed using Eq. (5.23), and $\mathbf{x}_{\Gamma,ik}$ the position of the interface Γ on a crossing pair ik, interpolated from the position of nodes i and k as in Eq. (2.16), recalled here for convenience:

$$\mathbf{x}_{\Gamma,ik} = \theta \mathbf{x}_k + (1 - \theta) \mathbf{x}_i \tag{5.26}$$

5.2.2 Conversion criteria between Eulerian and Lagrangian frameworks

5.2.2.1 Eulerian-to-Lagrangian transfer

In order to meet the sphericity condition when transforming an Eulerian droplet into a Lagrangian Point Particle (LPP), the droplet of interest needs to be small enough so that surface tension remains the dominant force, and thus maintains the spherical shape of the droplet. To this aim, for instance, a criterion based on the critical Weber number $\mathrm{We}_{\text{crit}}$, at which droplets begin to break up, can be employed [1]. Nonetheless, developing generic physical-based criteria is challenging, and is left for future improvements of the proposed conversion methodology. In this work, as widely done in the literature for cartesian grids, we propose to use criteria mainly based on numerical and geometrical considerations, adapted here to unstructured grids.

5.2.2.1.1 Size criterion - Threshold volumes

Size criteria are usually defined based on an effective resolution of the droplet (i.e. the number of grid points in the drop diameter). The idea behind this definition is that a poorly-resolved Eulerian droplet - less than 2 to 5 points in the diameter according to [199] - will not be properly represented on the Eulerian grid, and in the context of the ACLS method, this can lead to mass losses: a very small liquid fragment (e.g. resolved with one or two points) advected over a long distance may eventually vanish. A threshold volume V_{cut} can thus be defined as [25, 27, 1]:

$$V_{\mathrm{cut}} = \frac{\pi d_{\mathrm{cut}}^3}{6};\ d_{\mathrm{cut}} = \alpha \Delta x \tag{5.27}$$

with α the number of points in the cut diameter.

In this work, inspired by the idea of Zuzio et al. (2017) [28] which consists in discriminating small and medium liquid structures, two threshold volumes are used: a small and a medium threshold. They are defined respectively as:

$$V_{\mathrm{cut,small}} = \frac{\pi d_{\mathrm{cut,small}}^3}{6};\ d_{\mathrm{cut}} = \alpha_{\mathrm{small}} \Delta x \tag{5.28}$$

and

$$V_{\mathrm{cut,medium}} = \frac{\pi d_{\mathrm{cut,medium}}^3}{6};\ d_{\mathrm{cut}} = \alpha_{\mathrm{medium}} \Delta x \tag{5.29}$$

where α_{small} and α_{medium} are user-defined parameters. Typically, $\alpha_{\mathrm{small}} = 2$ and $\alpha_{\mathrm{medium}} = 10$.

We call "small liquid inclusions" the droplets for which $V_d < V_{\mathrm{cut,small}}$, and "medium liquid inclusions" the droplets for which $V_{\mathrm{cut,small}} < V_d < V_{\mathrm{cut,medium}}$. A small liquid inclusion is converted into an LPP without further considerations, whereas medium liquid inclusions will undergo other conversion criteria to determine whether they should be treated as Lagrangian particles or not.

5.2.2.1.2 Shape criterion - Aspect ratio

For medium liquid inclusions, a shape criterion is to be verified. Indeed, a more resolved liquid structure can be more subject to shape variations, and thus a sphericity check is necessary to avoid conversions of ligaments which would meet the medium volume criterion. The aspect ratio γ_d is considered to this aim. As mentioned in Ling et al. (2015) [27], a shape criterion should be defined using a geometric tolerance, noted ε_γ. The shape criterion used in the present study is the one of Ling et al. (2015) [27], and writes:

$$|\gamma_d - 1| < \varepsilon_\gamma \tag{5.30}$$

Indeed, $\gamma_d = 1$ means that the droplet of interest is perfectly spherical. The geometric tolerance is typically set to $\varepsilon_\gamma = 0.1$.

5.2.2.1.3 Distance criterion

As in Ling et al. (2015) [27], the proposed model does not include the droplet impact on the interface, and thus a distance criterion is used to verify that the liquid structure considered for transfer is sufficiently far from the resolved liquid-gas interface. This distance is typically the droplet equivalent diameter d_d. The proposed methodology consists in flagging nodes as part of an "influence region" of the equivalent spherical droplet associated with the liquid structure of

interest, and check that no other liquid structure is present in this region. This spherical influence region has a radius $\mathcal{R}_d$, computed as:

$$\mathcal{R}_d = \beta \mathrm{d}_d + \delta_\psi \tag{5.31}$$

where $\delta_\psi = 4\varepsilon$ is the interface profile thickness, defined in Eq. (2.3), and β is a user-defined parameter, typically $\beta = 1$. It should be noted that this criterion assumes that the liquid structure evaluated is spherical. Hence, this criterion applies to all medium liquid inclusions which already verify the shape criterion. The influence region is depicted in Fig. 5.1. A node i is in the influence

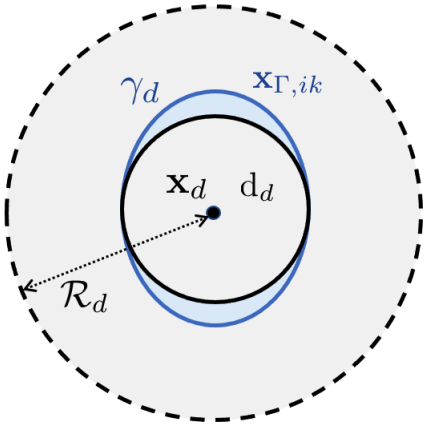


Figure 5.1: Illustration of the distance criterion, using the influence region (delimited by the black dashed line) of the equivalent spherical droplet (black solid line) associated with the resolved liquid structure of interest (blue solid line). The resolved droplet is represented as a large ellipsoid to ease the reading of the figure.

region if the relation

$$\phi_{d,i} = |\mathbf{x}_d - \mathbf{x}_i| < \mathcal{R}_d \tag{5.32}$$

is verified. This idea is inspired by the work of Zuzio et al. (2017) [28], which consists in associating a level set function to particles as part of a re-impact procedure of an LPP onto a resolved interface. Thus, the proposed algorithm can be partially re-used for the purpose of a Lagrangian-to-Eulerian transfer. As seen in Fig. 5.1, a robust criterion is required in order to avoid the self-detection of the liquid structure. The proposed criterion checks that only gas is present at the nodes (i.e. $\psi_i < 0.5$) belonging to the following region:

$$\max_{ik \in \mathrm{cp}} |\mathbf{x}_{\Gamma,ik} - \mathbf{x}_d| + \delta_\psi < |\mathbf{x}_d - \mathbf{x}_i| < \mathcal{R}_d \tag{5.33}$$

If this is the case, the medium liquid inclusion candidate is eventually qualified for transfer to the Lagrangian framework, as it verifies all criteria: size, shape, and distance.

5.2.2.2 Lagrangian-to-Eulerian transfer

In order to account for potential impacts of Lagrangian particles onto the resolved liquid-gas interface, a re-impact algorithm is necessary. Such algorithm is not implemented yet in YALES2 and is left for future work. Nevertheless, the principles of such algorithm are given in this section, according to the methodologies of Herrmann (2010) [25], and of Zuzio et al. (2017) [28], which

are suited for level-set interface-capturing methods. The first step is to determine whether a Lagrangian particle is on a collision course or not. For such detection, two main criteria can be used: a distance criterion [25, 28], and a direction criterion [28]. The distance criterion typically writes:

$$|\mathbf{x}_p - \mathbf{x}_i| < r_p + \Delta x \tag{5.34}$$

where $\mathbf{x}_p$ is the position of the particle, $\mathbf{x}_i$ are the coordinates of the node i, and r_p is the particle radius. As mentioned in the previous section, this equation strongly resembles to Eq. (5.32). The direction criterion is expressed as follows:

$$\mathbf{u}_p \cdot \boldsymbol{\nabla}\phi > 0 \tag{5.35}$$

with $\mathbf{u}_p$ the particle velocity, and ϕ the signed-distance function. If a particle impacts the resolved surface, the level set field then needs to be reconstructed as:

$$\phi_i = r_p - |\mathbf{x}_p - \mathbf{x}_i| \tag{5.36}$$

for all nodes inside the droplet. The reinitialization equation is then applied to update the level set field in the gaseous phase.

Please note that such methodology needs special attention to ensure mass and momentum conservation during transfer. Moreover, in our framework with AMR, it requires to suddenly and strongly refine the mesh locally, which is challenging. Thus, an implementation of such procedure might necessitate progressive grid refinement as the particle approaches the resolved interface, leading in turn to other challenges for two-way coupled Lagrangian Particle Tracking (LPT) methods.

5.2.3 Coupling with Adaptive Mesh Refinement

Inspired by the pioneering work of Tomar et al. (2010) [26] and by Zuzio et al. (2017) [28], the present multiscale procedure works in synergy with the AMR process. The idea is to coarsen the grid after conversion so that the newly introduced Lagrangian particle is much smaller than the local cell size. Even if only the one-way coupling strategy is used in this work, this synergy is developed in order to introduce a two-way coupling strategy in the near future.

Once a resolved droplet is qualified for transfer, i.e. meets all Euler-to-Lagrange conversion criteria, it first needs to be removed from the Eulerian grid. This operation is simply done by setting $\psi_i = 0$ for all nodes i belonging to the narrow band built around the droplet of interest. Then, the narrow band is rebuilt to account for the removal of interface. Eventually, the signed-distance function ϕ needs to be reconstructed, and hence the GPMM algorithm of Section 2.2.2 is employed again to obtain the new ϕ_{GPMM} field. This field is indeed necessary for the AMR process, as seen in Section 2.4. Once all these operations have been performed, the AMR process automatically coarsen the grid. In order to avoid AMR to be triggered at every iteration and to limit the overhead, the Eulerian-to-Lagrangian conversion is employed only when AMR is to be triggered anyway by the Eulerian solver. This way, no additional AMR process is called.

5.2.4 Summary of the conversion procedure

The present chapter, which details all the steps and conditions of the Euler-to-Lagrange conversion procedure, can be summarized in the flowchart presented in Fig. 5.2.

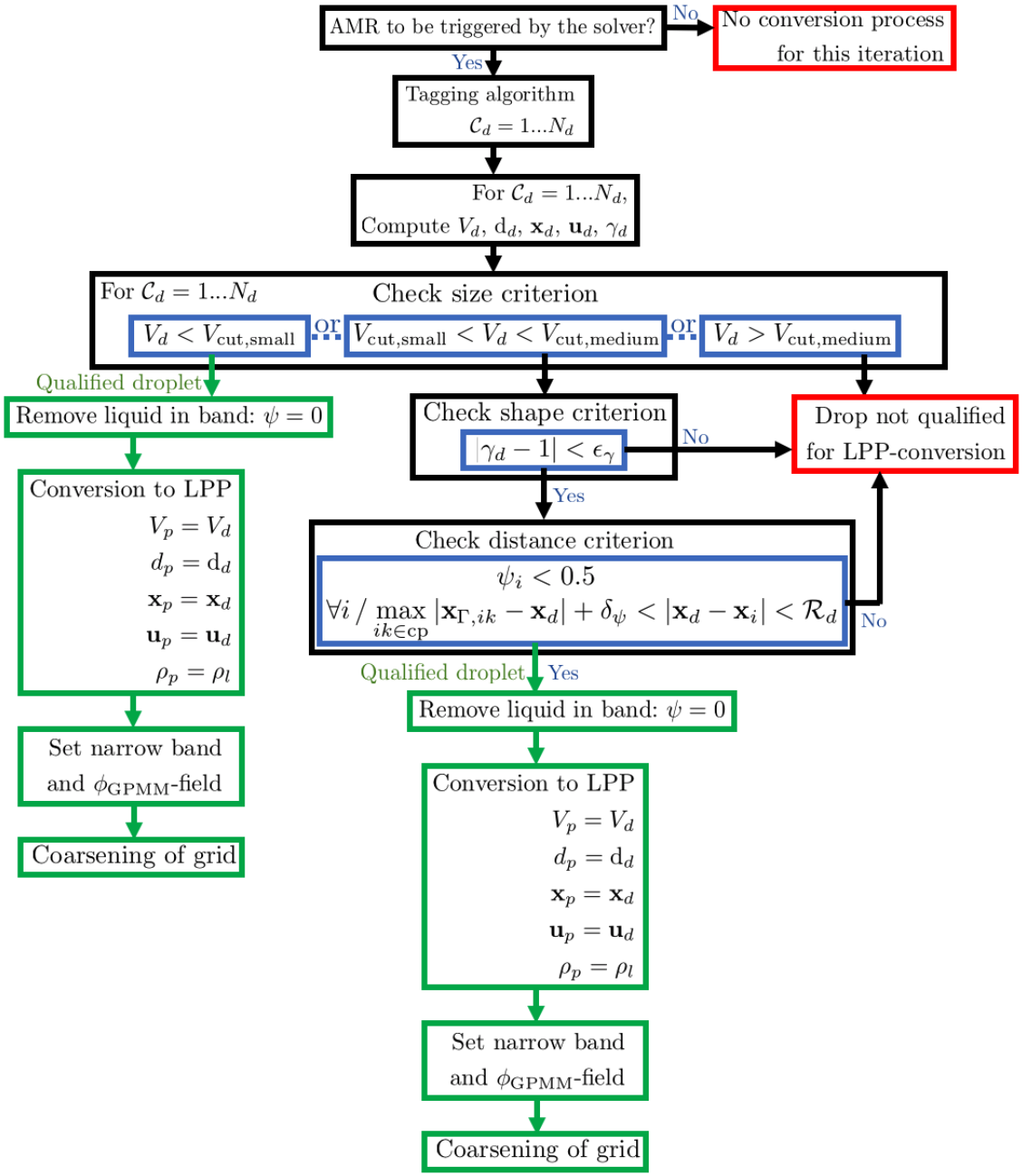


Figure 5.2: Flowchart of the Eulerian-to-Lagrangian conversion.

CHAPTER 6

Euler-Lagrange simulations using adaptive unstructured grids

Contents

6.1 Settling droplet 127
6.2 Accelerated droplet 129
6.3 High-pressure kerosene jet in crossflow 131
6.3.1 Visualization of the jet 131
6.3.2 Computational performances 136

This sixth chapter firstly aims to validate the hybrid multi-scale Euler-Lagrange procedure presented in the previous chapter, on canonical test cases using adaptive tetrahedral meshes. The Euler-Lagrange strategy is then applied to the same liquid jet in crossflow case than the one conducted in the first part of the manuscript (Chapter 4). As a consequence, the jet topologies and computational performances of the hybrid and fully Eulerian simulations are compared.

6.1 Settling droplet

As a first validation of the hybrid Eulerian-to-Lagrangian conversion and of the one-way coupled Lagrangian Particle Tracking solver itself, the well-known droplet settling test case is investigated. This study follows the ones of Ling et al. (2015) [27] and of Zuzio et al. (2017) [28].

A static droplet of diameter D is initialized in a cuboid tank of dimensions $L_x = L_z = L_y/2$, filled with a quiescent gas. The droplet falls along the $y < 0$ direction under the influence of gravity, with an increasing vertical velocity $|\mathrm{u}_y|$ over time, until the equilibrium between gravitational and drag forces is reached. At this point, the acceleration is zero and hence the velocity is constant. It is thus possible to extract a terminal settling velocity u_t from the simulation and compare it with existing empirical correlations.

The study of Ling et al. (2015) [27] reports a substantial retarding effect of the walls on the terminal falling velocity of the droplet. This effect depends on the ratio between the droplet or particle diameter $D = d_p$, and the smallest dimension of the cuboid gas tank $L_x = L_z$. In their work, where $d_p/L_x = 1/10$, the correlation of Di Felice (1996) [200] for a cylindrical tube is used to correct the drag calculation in the equation of motion of the particle:

$$\eta = \frac{\mathrm{u}_t}{\mathrm{u}_{t,\mathrm{unbounded}}} = \left(\frac{1 - d_p/L_x}{1 - 0.33 d_p/L_x} \right)^{2.7} \tag{6.1}$$

To limit these bound effects, the cuboid numerical domain in the present work is chosen so that $d_p/L_x = 1/50$. No correction of the drag calculation accounting for the walls is used in the present study.

As in Zuzio et al. (2017) [28], the Eulerian droplet is converted into a Lagrangian Point Particle (LPP) at the first iteration. The numerical setup of the present study is summarized in Fig. 6.1. The Eulerian droplet, initially placed at $(0, L_y/20, 0)$, is refined at its interface with a cell size $\Delta x_{\min}$ chosen so that the diameter of the droplet D contains 32 points: $D/\Delta x_{\min} = 32$. The droplet is detected and converted into an LPP of diameter $d_p = D$ at the end of the first iteration, and the coarsening procedure activates until $d_p/\Delta x_{\max} = 1/10$. Please note however that, as mentioned in Ling et al. (2015) [27] and in Zuzio et al. (2017) [28], the one-way coupled LPP simulation is mesh independant in this test case. The physical parameters are the ones of Ling et

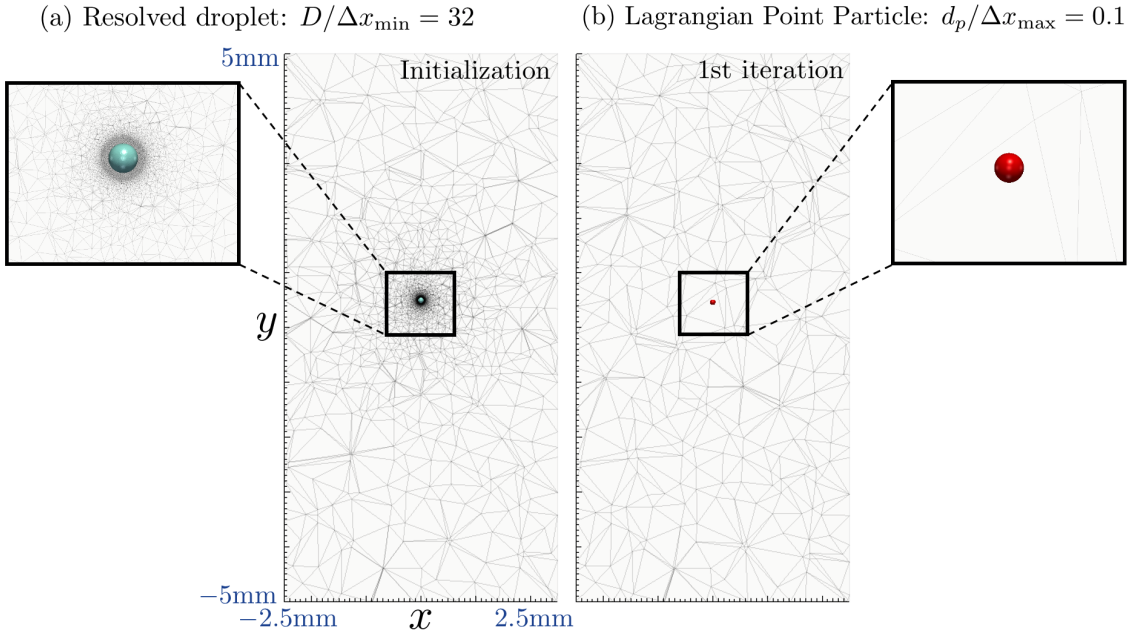


Figure 6.1: Numerical setup for the settling droplet test case: (a) At the initialization; (b) At the end of the first iteration. The droplet falls along the $y < 0$ direction.

al. (2015) [27], and are summarized in Table 6.1. The simulation time is $T = 0.03$s as in the previous studies mentioned. The resulting temporal evolutions of the particle vertical position and

Table 6.1: Physical parameters for the settling droplet test case.

$\rho_l\,(\mathrm{kg.m^{-3}})$	$\rho_g\,(\mathrm{kg.m^{-3}})$	$\mu_l\,(\mathrm{kg.m^{-1}.s^{-1}})$	$\mu_g\,(\mathrm{kg.m^{-1}.s^{-1}})$	$d_p\,(\mathrm{m})$	$\sigma\,(\mathrm{N.m^{-1}})$	$\mathrm{g}\,(\mathrm{m.s^{-2}})$
1000	10	10^{-3}	10^{-4}	10^{-4}	0	9.8

velocity are displayed in Fig. 6.2 (a) and (b), respectively. The particle falling occurs as expected for both of these values: the temporal evolutions correspond to what has been presented in Ling et al. (2015) [27] and in Zuzio et al. (2017) [28]. The terminal velocity can be compared to the empirical correlation of Clift and Gauvin (1970) [201], which is obtained through the standard drag calculation for unbounded domains. As seen in Fig. 6.2 (b), the simulation value agrees well with the empirical value. To quantify the result, the relative error is presented in Table 6.2.

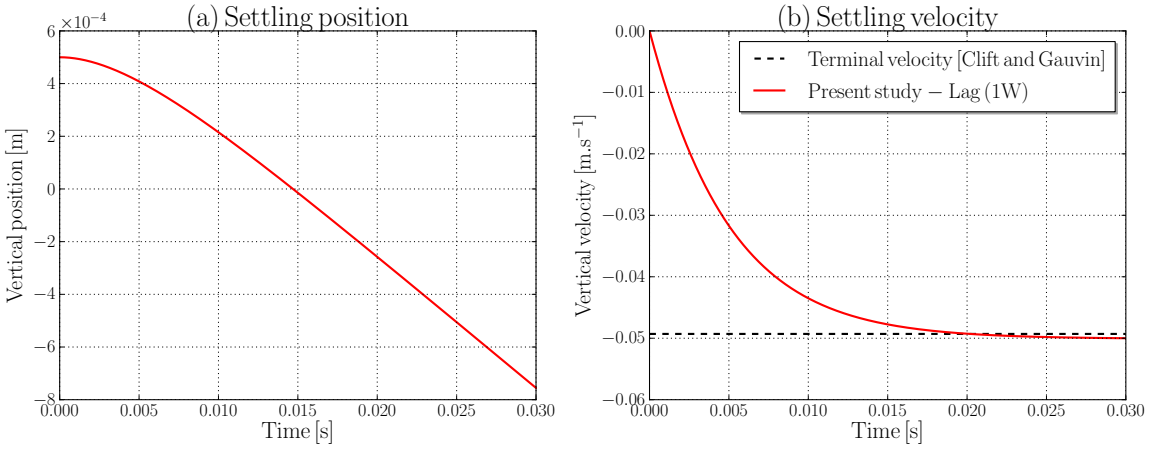


Figure 6.2: (a) Temporal evolution of the y-position of the falling droplet; (b) Temporal evolution of the u_y-velocity of the falling droplet against the terminal velocity for unbounded domains of Clift and Gauvin (1970) [201].

Table 6.2: Terminal velocity result and comparison for the settling droplet test case.

Computed $\mathrm{u}_t\,(\mathrm{m.s}^{-1})$	Terminal velocity of Clift and Gauvin [201] $\mathrm{u}_{t,\mathrm{CG}}\,(\mathrm{m.s}^{-1})$	Relative error
0.0500	0.0493	+1.42%

6.2 Accelerated droplet

In order to test the robustness of the Eulerian-to-Lagrangian conversion/coarsening procedure, the accelerated droplet test case of Zuzio et al. (2017) [28] is then run for validation. A spherical liquid droplet of diameter $D = 0.1\mathrm{m}$, initially at rest at the center of a cubic box of dimension $L = 6.5\mathrm{m}$, is accelerated along the $x > 0$ direction by a uniform axial air flow of velocity $\mathbf{u}(t = 0) = (U_0, 0, 0)$. As mentioned in their work, this physical situation is relevant for atomization simulations, as it corresponds to droplet generation configurations. Again, the one-way coupling approximation is used in this test case.

The physical parameters are the ones of Zuzio et al. (2017) [28] and are presented in Table 6.3. These physical parameters yield a Weber number of $\mathrm{We}_D = (\rho_l U_0^2 D)/\sigma = 2$ and a Reynolds

Table 6.3: Physical parameters for the accelerated droplet test case.

$\rho_l\,(\mathrm{kg.m}^{-3})$	$\rho_g\,(\mathrm{kg.m}^{-3})$	$\mu_l\,(\mathrm{kg.m}^{-1}.\mathrm{s}^{-1})$	$\mu_g\,(\mathrm{kg.m}^{-1}.\mathrm{s}^{-1})$	$D = d_p\,(\mathrm{m})$	$\sigma\,(\mathrm{N.m}^{-1})$	$U_0\,(\mathrm{m.s}^{-1})$
2	1	10^{-3}	10^{-3}	0.1	10^{-3}	0.1

number of $\mathrm{Re}_D = (\rho_l U_0 D)/\mu_l = 10$. The acceleration is studied up to a final time $T = 2\mathrm{s}$, and the time step is set to $\Delta t = 10^{-3}\mathrm{s}$ for all simulations. Slip conditions are applied for the walls tangent to the air flow, and inlet/outlet boundary conditions for walls perpendicular to it.

Five simulations are run, four fully Eulerian simulations of increasing resolutions (4, 8, 16, 32 points in the drop diameter), and one LPP simulation initialized using the finest grid (32 points in the drop diameter at initialization). For the LPP simulation, as in the settling droplet study, the

droplet is detected and converted into a Lagrangian Point Particle at the end of the first iteration. The coarsening procedure then activates so that $d_p/\Delta x_{\max} = 0.1$. The numerical setup for the LPP simulation is displayed in Fig. 6.3.

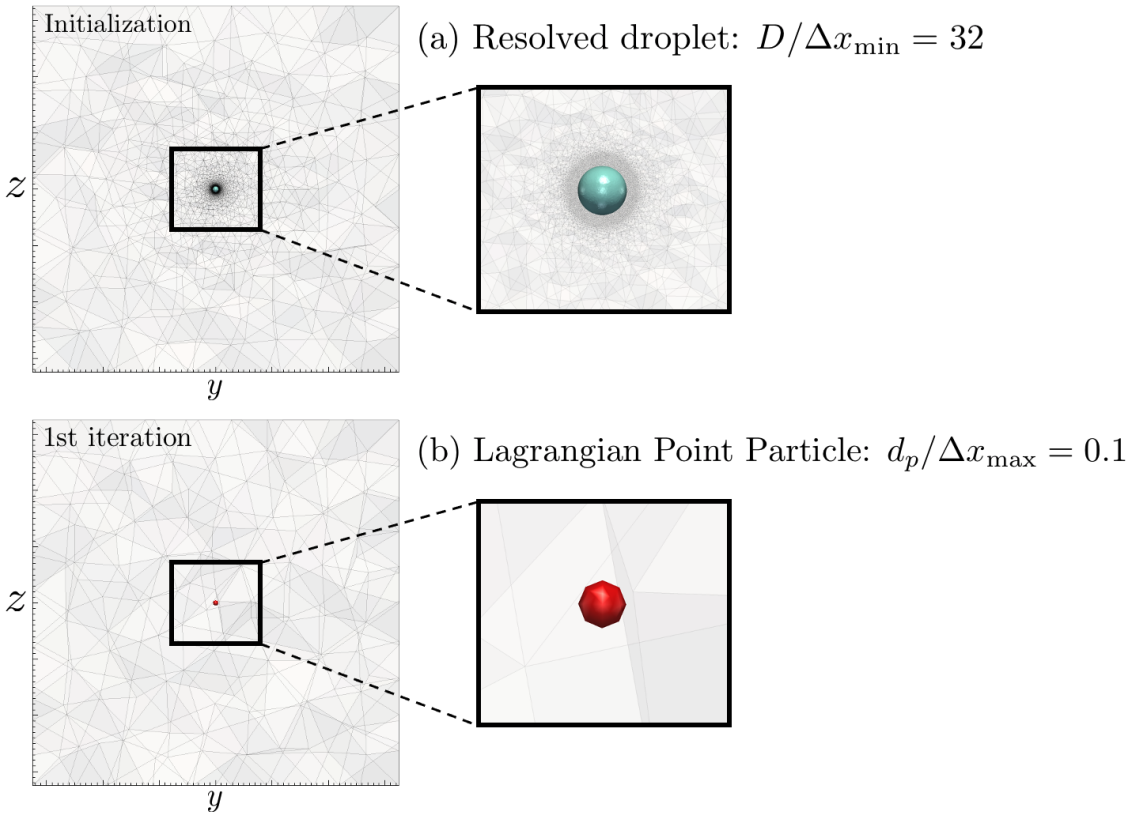


Figure 6.3: Numerical setup for the accelerated droplet test case (LPP simulation): (a) At the initialization; (b) At the end of the first iteration. The droplet is accelerated along the $x > 0$ direction.

The position $\mathbf{x}_d$ and velocity $\mathbf{u}_d$ of the Eulerian droplet are calculated according to Eqs. (5.23) and (5.24), respectively.

The temporal evolution of the droplet/particle position and velocity for all simulations are shown in Fig. 6.4 (a) and (b), respectively. The final values are reported in Table 6.4. Spatial convergence of the Eulerian simulations is observed, demonstrating the consistency of the averaging procedure to calculate Eulerian droplet characteristics. The finest Eulerian simulation can be considered as a reference solution. The LPP simulation underestimates the acceleration of the droplet, nonetheless the behavior is correct overall despite a strong coarsening of the mesh. It should be noted that the LPP simulation is sensitive to the initial position of the Lagrangian particle on the grid at conversion, due to interpolation errors of the gas velocity at the particle. Table 6.4 also exhibits the final number of tetrahedral mesh cells for each conducted simulation. The results demonstrate an approximate gain of factor 20 on the grid size using the Lagrangian model compared to the coarsest Eulerian simulation, and a gain of factor 170 compared to the finest Eulerian simulation.

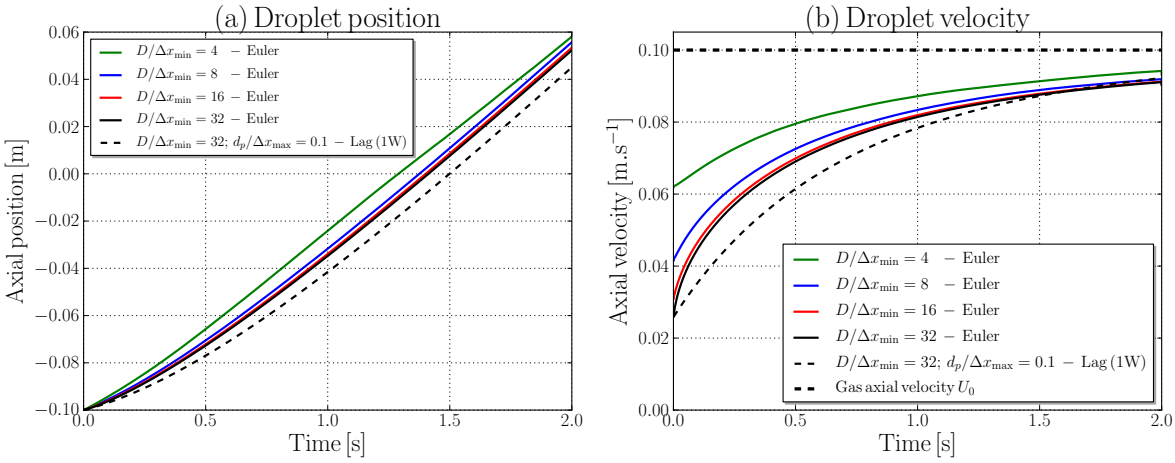


Figure 6.4: (a) Temporal evolution of the x-position of the accelerated droplet; (b) Temporal evolution of the u_x-velocity of the accelerated droplet.

Table 6.4: Final axial positions x_f, axial velocities $\mathrm{u}_{x,f}$, and number of grid cells $N_{\mathrm{cells},f}$ for the accelerated droplet test case.

Simulation	x_f (m)	$\mathrm{u}_{x,f}$ (m.s^{-1})	$N_{\mathrm{cells},f}$
Euler 4	0.0581	0.0932	76, 772
Euler 8	0.0557	0.0911	120, 319
Euler 16	0.0534	0.0904	240, 039
Euler 32	0.0523	0.0904	679, 423
Lagrange (1W)	0.0451	0.0922	3, 993

6.3 High-pressure kerosene jet in crossflow

The final application case of this study is a hybrid Eulerian-Lagrangian high-pressure kerosene jet in crossflow, already presented for the fully Eulerian case in Section 4.2. The operating point corresponding to the highest gaseous Weber number $\mathrm{We}_g = 1470$ is chosen, for its ability to produce more droplets and the computational challenges that it poses. This simulation is thus well-suited to assess the computational performances of the proposed multiscale strategy, and compare them against the fully Eulerian case. As in Section 4.2, two interface resolutions are studied, namely $\Delta x_{\min} = 20\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 22.5$) and $\Delta x_{\min} = 10\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 45$). The boundary condition for Lagrangian particles is a slip-on-wall condition: $\mathbf{u}_p \cdot \mathbf{n}_{\mathrm{wall}} = 0$ (only the tangential component to the wall is non-zero). In order to perform a large number of conversions, the cut diameter for medium liquid inclusions is set to $16\Delta x_{\min}$, and the geometric tolerance ε_γ is set to 15%, for both resolutions.

6.3.1 Visualization of the jet

A comparison of jet topologies between the hybrid Eulerian-Lagrangian simulation and the fully Eulerian one for the coarse resolution $\Delta x_{\min} = 20\mu\mathrm{m}$ is shown in Fig. 6.5, at $t = 0.3\mathrm{ms}$ as in Section 4.2. For the multiscale simulation, the Lagrangian Point Particles are also represented.

As expected, much more droplets/particles are present in the domain for the hybrid simulation. Atomization and column instabilities are seen to be more pronounced when using the one-way coupled Lagrangian Particle Tracking, resulting in an upward shift of the instantaneous jet trajectory. Eventually, the side view displays a good agreement between the two simulations in the impact region of drops/particles onto the bottom wall, confirming the choice of a slip-on-wall boundary condition.

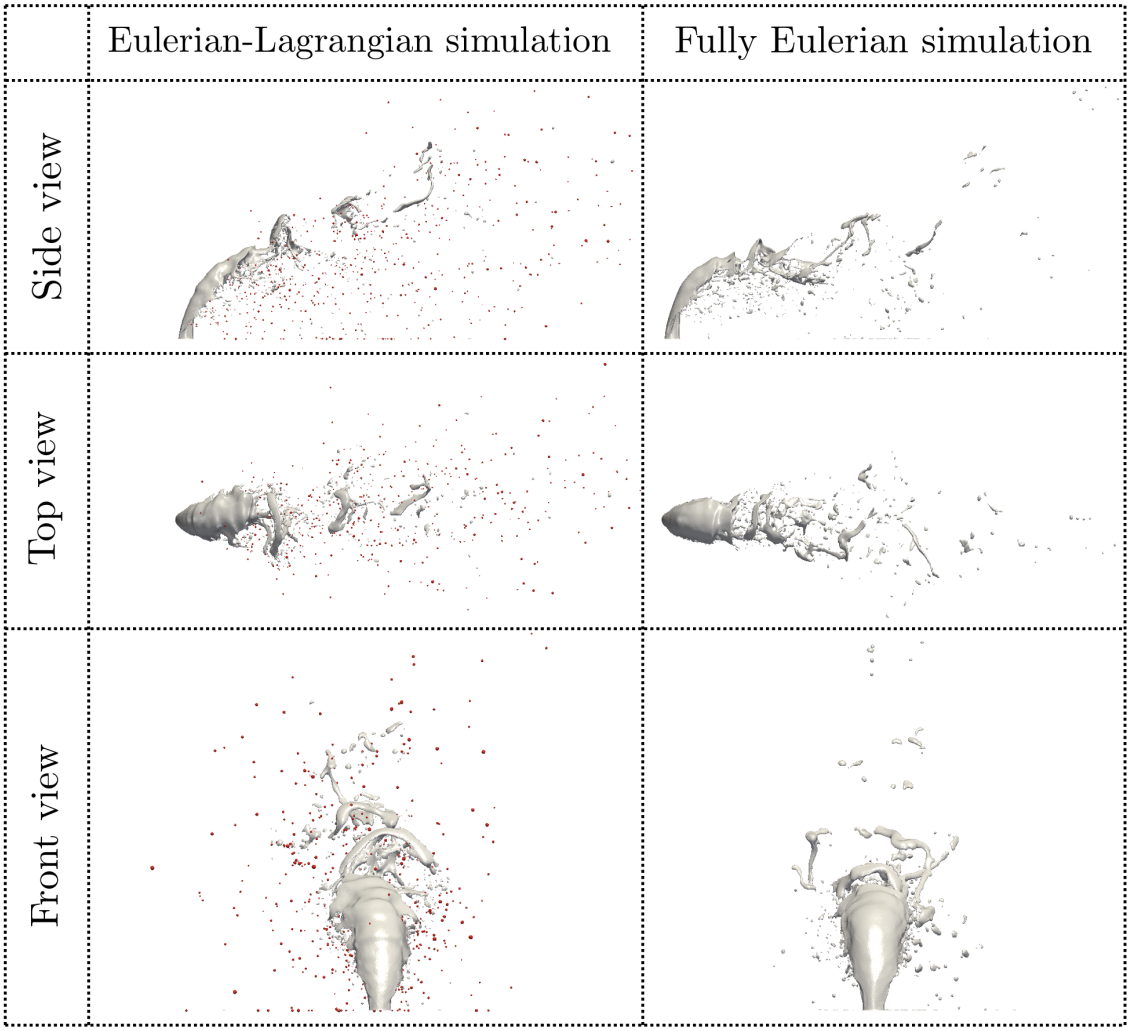


Figure 6.5: Topology of the high-pressure kerosene jet in crossflow at $t = 0.3\text{ms}$, using the hybrid Euler-Lagrange strategy versus the fully Eulerian simulation of [180]. $\text{We}_g = 1470$; $\Delta x_{\min} = 20\mu\text{m}$; $d_{\text{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$.

A more developed hybrid jet for the coarse resolution $\Delta x_{\min} = 20\mu\text{m}$, at $t = 0.375\text{ms}$, is displayed in Figs. 6.6 and 6.7. Over 1000 particles are present in the domain at this time instant. It should be noted that the one-way coupling approximation is less relevant in the near-core region. Indeed, the mass fraction of the particles is higher in this region of the flow, and thus the need of a two-way coupled Lagrangian strategy together with a re-impact algorithm might arise to properly simulate the jet instabilities. Figs. 6.6 and 6.7 also show that a large number of particles impacts the bottom wall. This is particularly true when the surface breakup mechanism is dominant [41], which is the case for these operating conditions. Hence, a film model could be integrated in the hybrid strategy to improve the prediction of the spray behavior close to the bottom wall. The

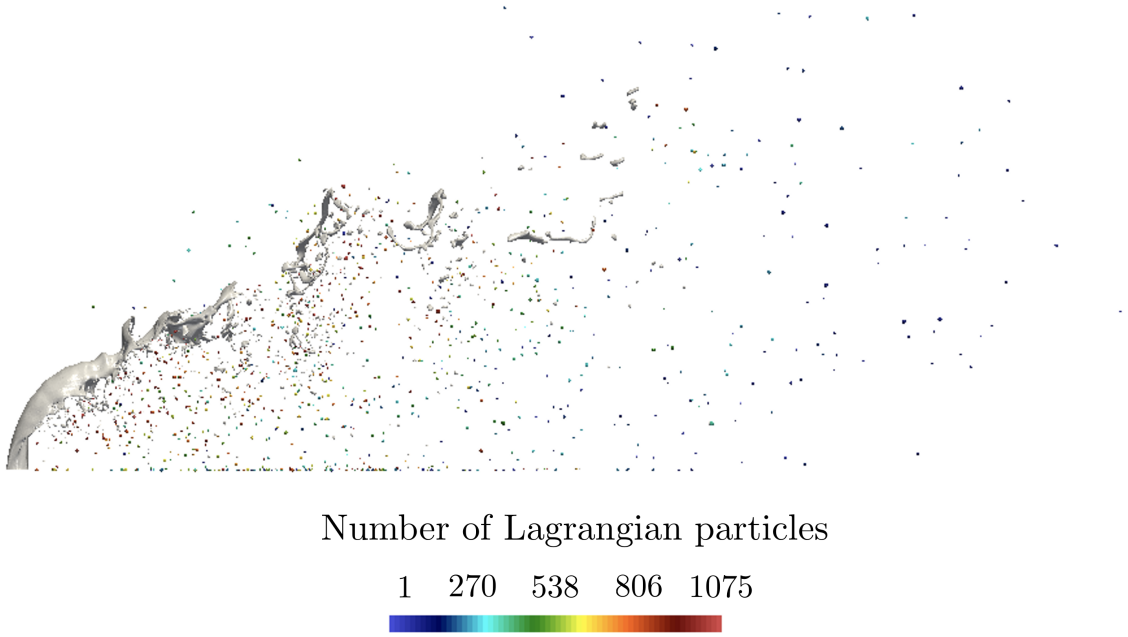


Figure 6.6: Topology of the high-pressure kerosene jet in crossflow and count of Lagrangian particles at $t = 0.375\text{ms}$, using the hybrid Euler-Lagrange strategy. $\text{We}_g = 1470$; $\Delta x_{\text{min}} = 20\mu\text{m}$; $d_{\text{cut,medium}} = 16\Delta x_{\text{min}}$; $\varepsilon_\gamma = 0.15$.

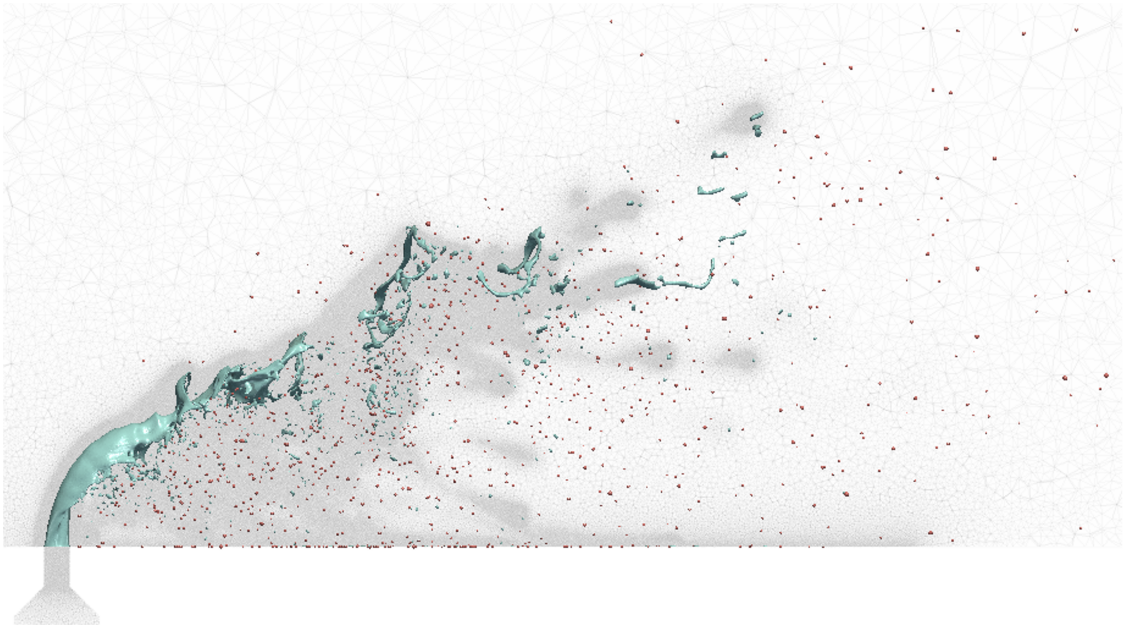


Figure 6.7: Topology of the high-pressure kerosene jet in crossflow along with the adaptive tetrahedral grid at $t = 0.375\text{ms}$, using the hybrid Euler-Lagrange strategy. $\text{We}_g = 1470$; $\Delta x_{\text{min}} = 20\mu\text{m}$; $d_{\text{cut,medium}} = 16\Delta x_{\text{min}}$; $\varepsilon_\gamma = 0.15$.

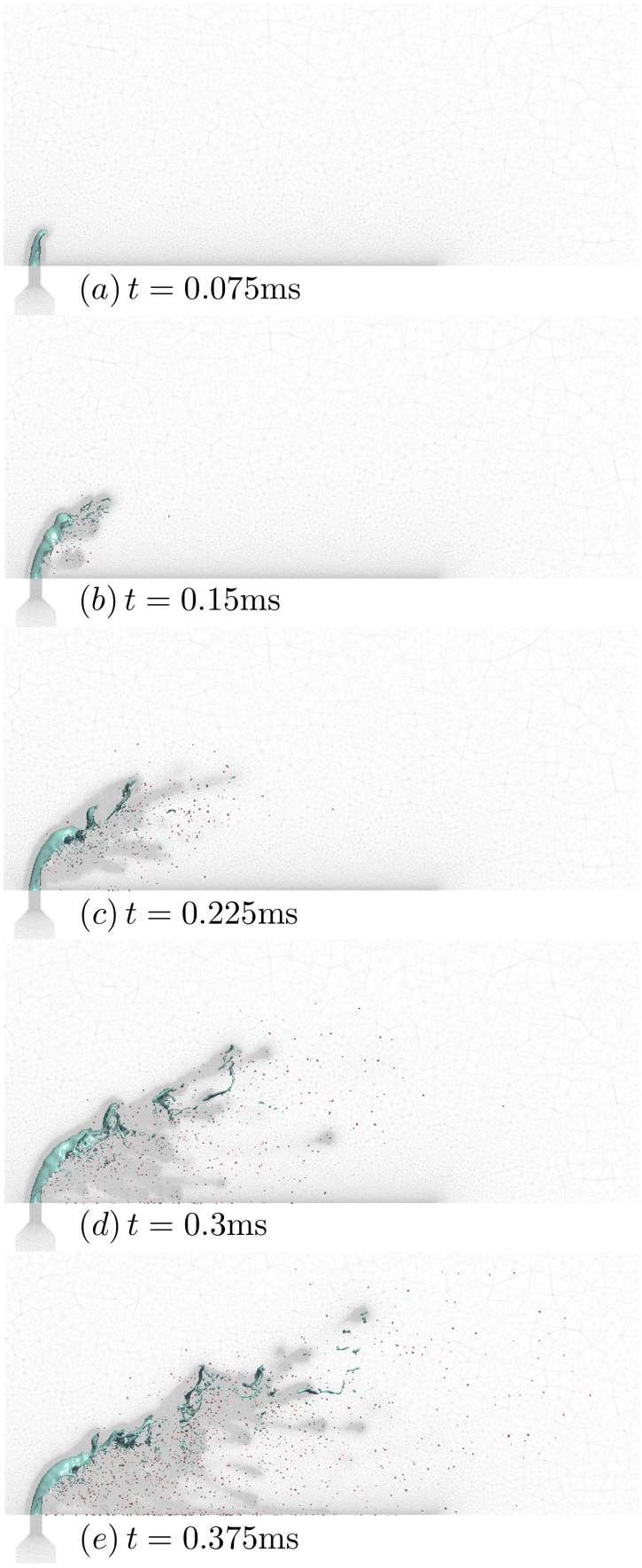


Figure 6.8: Temporal evolution of the high-pressure kerosene jet in crossflow along with the adaptive tetrahedral grid, using the hybrid Euler-Lagrange strategy. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu$m; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$.

temporal evolution of the interface, still for the coarse resolution $\Delta x_{\min} = 20\mu\text{m}$, along with the particles and the adaptive tetrahedral grid is shown in Fig. 6.8. It is clearly observed that the particles travel much further than an equivalent resolved Eulerian droplet would, due to mass losses induced by the poorly-resolved drop disappearance after a certain distance of advection.

The particle size distribution at time instant $t = 0.375\text{ms}$ is shown in Fig. 6.9. All particles with a diameter $d_p < 2\Delta x_{\min} = 40\mu\text{m}$ are issued from converted droplets based on their size only, without neither shape nor distance considerations (small liquid inclusions). The rest of the particles (medium liquid inclusions: $2\Delta x_{\min} = 40\mu\text{m} < d_p < 16\Delta x_{\min} = 320\mu\text{m}$) are issued from drop conversions based on size, shape, and distance criteria. As observed in the histogram, no particle with $d_p > 4.5\Delta x_{\min} = 90\mu\text{m}$ are present in the domain, meaning that the medium size criterion could be more restrictive without changing the actual distribution. To quantitatively assess the typical droplet size in the present study, and compare it to the experimental study of Becker and Hassa (2002) [41], the Sauter Mean Diameter (SMD) d_{32} can be computed as [202]:

$$d_{32} = \frac{\sum_{p=1}^{N_p} d_p^3}{\sum_{p=1}^{N_p} d_p^2} \tag{6.2}$$

which yields $d_{32} = 50.2\mu\text{m}$ for the present particle distribution. The experiment of Becker and Hassa (2002) [41] reports an SMD far downstream the injection location (plane $x = 80\text{mm}$) of $d_{32,\text{exp}} = 30.8\mu\text{m}$, which is significantly lower than in the present simulation. Thus, secondary atomization models would be needed for Lagrangian particle breakup, to better match the experimental data using this interface cell size of $\Delta x_{\min} = 20\mu\text{m}$.

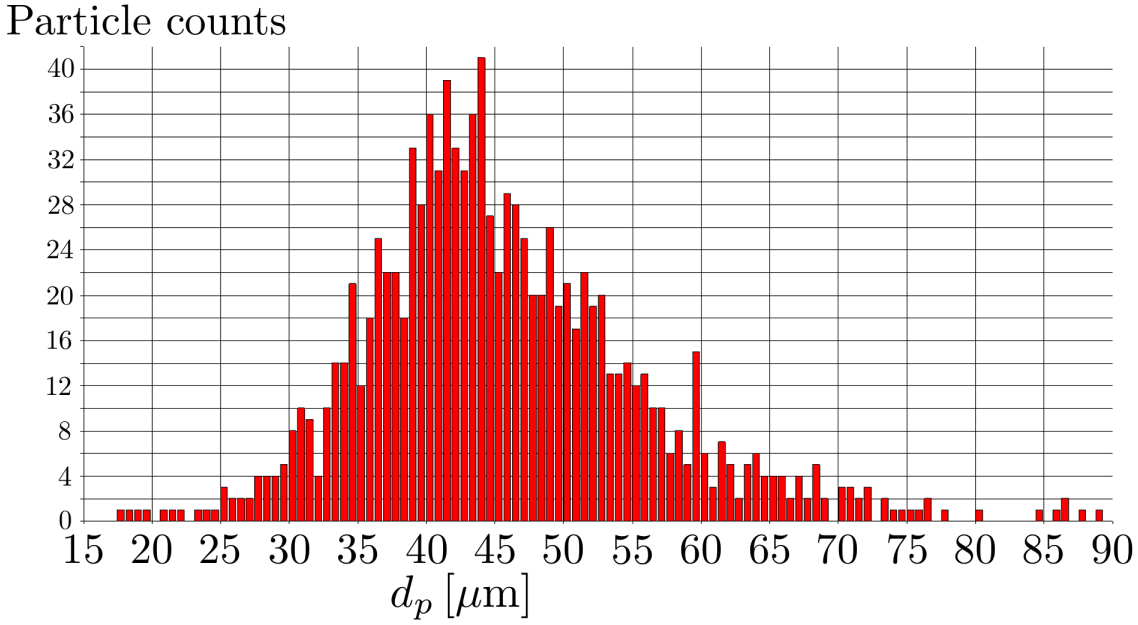


Figure 6.9: Particle diameter distribution at $t = 0.375\text{ms}$, using the hybrid Euler-Lagrange strategy. $\text{We}_g = 1470$; $\Delta x_{\min} = 20\mu\text{m}$; $d_{\text{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$.

An early stage of the jet contour with particles at the fine resolution $\Delta x_{\min} = 10\mu\text{m}$, at $t = 0.15\text{ms}$, is shown in Fig. 6.10. The interface contour is cut by the mid-plane, on which the adaptive tetrahedral grid is represented for Fig. 6.10 (a), and the velocity field for Fig. 6.10 (b). Eventually, a closeup on a Lagrangian conversion is provided in Fig. 6.11.

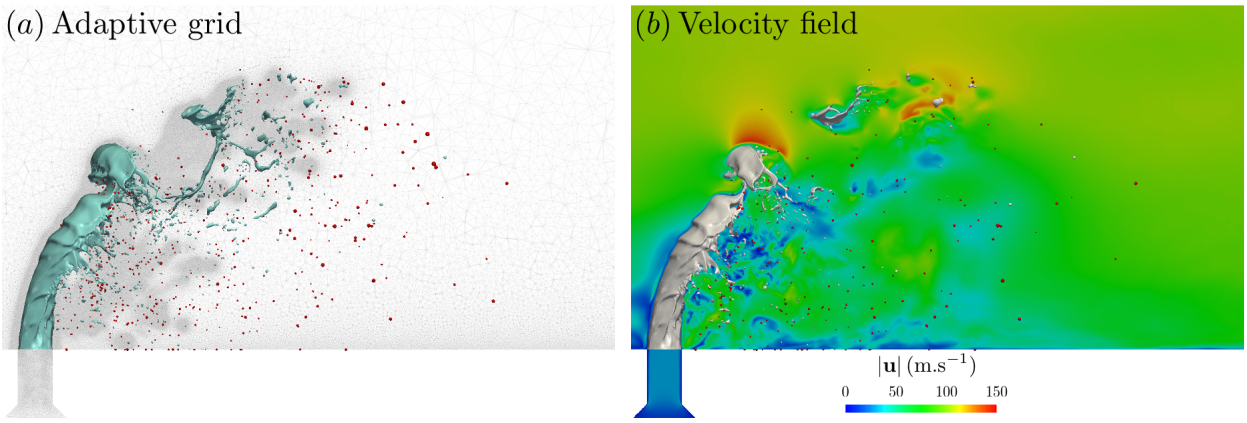


Figure 6.10: Early stage of the development of the Eulerian-Lagrangian high-pressure kerosene jet in crossflow along with (a) the adaptive tetrahedral grid, and (b) the velocity field in the mid-plane. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$.

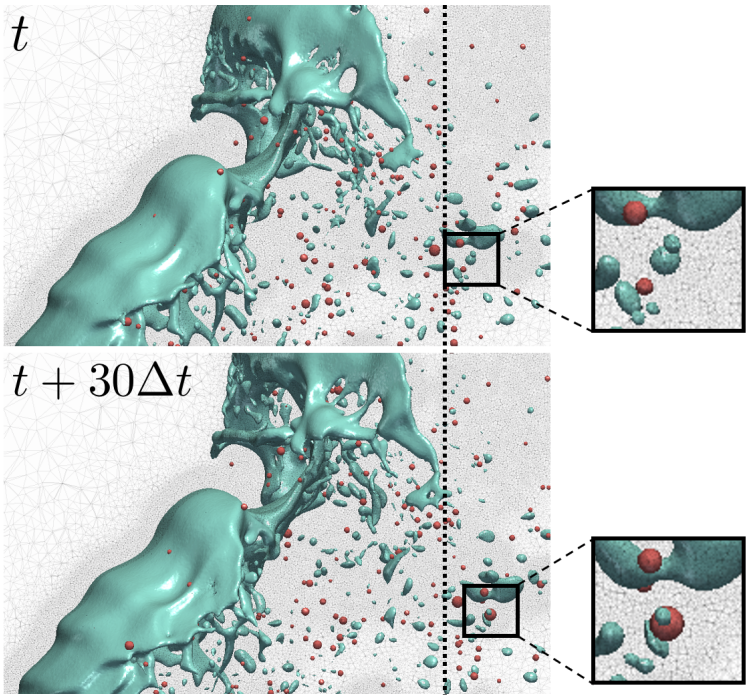


Figure 6.11: High-pressure kerosene jet in crossflow - Closeup on an Euler-to-Lagrange conversion. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu\mathrm{m}$; $d_{\mathrm{cut,medium}} = 16\Delta x_{\min}$; $\varepsilon_\gamma = 0.15$.

6.3.2 Computational performances

The simulation features for the coarse simulation $\Delta x_{\min} = 20\mu\mathrm{m}$ are given in Table 6.5, for both the Eulerian and the hybrid Eulerian-Lagrangian cases. The values correspond to time instant $t = 0.3\mathrm{ms}$. The number of droplets, resolved and modeled, is displayed. It is seen that the Euler-Lagrange simulation almost doubles the number of drops/particles in the domain at this time instant compared to the Eulerian case, for less resolved liquid structures. As a consequence, the hybrid case employs a lighter mesh: 26M cells less (-19%). Similar mesh element loads per core are employed between the two simulations and thus their performances based on the

Reduced Computational Times (RCT; Eq. (3.8)), shown in Table 6.6, can be compared reliably. A gain of factor 2 in total RCT is displayed when performing the hybrid simulation. Indeed, as less resolved structures are present in the domain, the ACLS procedure for interface capturing and the Poisson linear solver are substantially faster. For the hybrid simulation, $\mathrm{RCT}_{\mathrm{Eul.-Lag.}}$ includes the conversion routines (drop tagging algorithm, checks of conversion criteria, removal of qualified liquid structures from the grid...) and particle transport. In particular, it includes the time spent in the routines which compute the droplet characteristics (volume, position, velocity, aspect ratio) and the additional call to the GPMM distance algorithm. Thus, the overhead induced by these additional calls is amply balanced by the smallest liquid-gas surface density (and hence the lighter grid). Moreover, converting the smallest resolved drops into Lagrangian particles, which can often be the fastest liquid structures in the domain for this case, avoids to remesh automatically too frequently because of these fast small droplets. As a consequence, it leads to less time spent in the AMR routine, even with the introduction of strong coarsening ratios. It should also be noted that as less cores are used for the hybrid simulation, the AMR procedure converges faster on average, due to less bad quality cells at core boundaries: this phenomenon has been detailed in Section 3.2.4. Eventually, the cost of the two simulations is given, with an advantage for the hybrid simulation.

Table 6.5: Number of computational cores, mesh cells and droplets generated for the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180], whose performance values are displayed in Chapter 4, Table 4.4. All values correspond to time instant $t = 0.3$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 22.5$).

Case	N_{cores}	N_{cells}	$N_{\mathrm{cells}}/N_{\mathrm{cores}}$	$N_{\mathrm{drops,resolved}}$	$N_{\mathrm{drops,modeled}}$	$N_{\mathrm{drops,total}}$
Euler	1024	138M	135K	596	0	596
Euler-Lagrange	720	112M	156K	550	474	1024

Table 6.6: Computational performances of the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180], whose performance values are displayed in Chapter 4, Table 4.5. All values correspond to time instant $t = 0.3$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 20\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 22.5$).

Case	RCT [μs]	$\mathrm{RCT}_{\mathrm{ACLS}}$ [μs]	$\mathrm{RCT}_{\mathrm{AMR}}$ [μs]	$\mathrm{RCT}_{\mathrm{Poisson}}$ [μs]	$\mathrm{RCT}_{\mathrm{Eul.-Lag.}}$ [μs]	CPU time [h]/Phys. time [ms]
Euler	1061.5	145.3	838.0	78.2	0	28,700
Eul.-Lag.	551.2	95.0	412.9	34.7	8.6	26,552

As for the coarser jets, Tables 6.7 and 6.8 present the computational informations relative to the fine Eulerian and hybrid Eulerian-Lagrangian cases at $\Delta x_{\min} = 10\mu\mathrm{m}$ ($d_{\mathrm{inj}}/\Delta x_{\min} = 45$), at a time instant closer to the beginning of injection: $t = 0.15$ms. Again, more droplets/particles are present in the domain for the hybrid simulation compared to the Eulerian one. The smaller increase in $N_{\mathrm{drops,total}}$ between the Eulerian and the hybrid cases ($+40\%$), compared to the coarse simulations ($+70\%$), is due to the earlier time instant studied. Nonetheless, a clear reduction in the number of resolved liquid structures and hence a lighter grid are already observed for the multi-scale simulation (-44M cells; -27%). The number of elements per core is significantly different between the two simulations, and hence it is not easy to clearly compare the RCT values. How-

ever, one can observe in Table 6.8 similar trends than for the coarse study regarding $\mathrm{RCT}_{\mathrm{ACLS}}$, $\mathrm{RCT}_{\mathrm{Poisson}}$, and $\mathrm{RCT}_{\mathrm{Eul.-Lag.}}$, with a doubled number of cores for the multiscale simulation. The large increase in $\mathrm{RCT}_{\mathrm{AMR}}$ and in cost can be explained by two main reasons: (i) The number of cores is too large for the beginning of the simulation, leading in particular to bad quality cells at the interface between processors keeping the AMR process from converging fast; (ii) the jet is not developed and atomized enough at this time instant for the conversion procedure to reach its full potential. Indeed, as seen in Fig. 6.8 (b) and (d) for the coarse case, many droplets are produced by the atomization process between $t = 0.15$ms and $t = 0.3$ms.

Table 6.7: Number of computational cores, mesh cells and droplets generated for the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180]. All values correspond to time instant $t = 0.15$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu$m ($d_{\mathrm{inj}}/\Delta x_{\min} = 45$).

Case	N_{cores}	N_{cells}	$N_{\mathrm{cells}}/N_{\mathrm{cores}}$	$N_{\mathrm{drops,resolved}}$	$N_{\mathrm{drops,modeled}}$	$N_{\mathrm{drops,total}}$
Euler	1152	163M	142K	850	0	850
Euler-Lagrange	1440	119M	83K	738	448	1186

Table 6.8: Computational performances of the hybrid Eulerian-Lagrangian jet in crossflow simulation and comparison to the fully-Eulerian simulation of [180]. All values correspond to time instant $t = 0.15$ms. $\mathrm{We}_g = 1470$; $\Delta x_{\min} = 10\mu$m ($d_{\mathrm{inj}}/\Delta x_{\min} = 45$).

Case	RCT [μs]	$\mathrm{RCT}_{\mathrm{ACLS}}$ [μs]	$\mathrm{RCT}_{\mathrm{AMR}}$ [μs]	$\mathrm{RCT}_{\mathrm{Poisson}}$ [μs]	$\mathrm{RCT}_{\mathrm{Eul.-Lag.}}$ [μs]	CPU time [h]/Phys. time [ms]
Euler	1274.8	151.0	1043.3	80.5	0	62,922
Eul.-Lag.	1375.3	127.4	1171.4	65.9	10.6	96,440

Conclusion and future work

Unstructured ACLS/GFM interface-capturing method for LES of primary atomization

An efficient, massively-parallel, and robust complete procedure for the computation of two-phase liquid-gas flows on unstructured meshes has been presented. Spatial convergence and accuracy of the interface-capturing technique, which includes the first unstructured version of the latest ACLS reinitialization equation, have been first demonstrated through well-known interface transport tests, both in 2D and 3D, and confirmed by the results obtained on canonical two-phase flow examples. Capillary-driven flows are correctly computed on triangular and tetrahedral grids, despite the challenges that poses the use of these irregular meshes. Complex 3D turbulent atomizing jets have been simulated, such as the jet issued from a triple-disk injector (Fig. 6.12), with a good prediction of their overall dynamics.

Figure 6.12: Volume rendering of the atomizing jet issued from a low-pressure compound nozzle.

The accuracy and convergence of the numerical framework presented in this manuscript can be further improved by using the higher-order finite-volume schemes of [83], which have proven to upgrade the operator discretization on unstructured grids and hence the interface curvature computation. The treatment of the viscous term in the momentum equation, which is not yet written in conservative form, and especially at the interface can also be upgraded, to be able to perform more accurate simulations where viscosity plays a central role, for instance computations at low Reynolds or Laplace numbers (e.g. rising bubble, settling droplet). Moreover, dedicated studies for the solver to achieve well-balanced and momentum-conservation properties need to be conducted (e.g. spurious currents relatively to a dissipation time scale for static and translated droplets [52], deformation of a convected high-density droplet [203], kinetic energy conservation of an oscillated droplet [203]). A revised, less restrictive, explicit or even an implicit treatment

of surface tension would help to alleviate the strong capillary time-step constraint, even if for the latter the removal of this constraint is not guaranteed [164]. Eventually, an additional subgrid LES model dedicated to primary atomization could also be integrated [204].

Adaptive Mesh Refinement (AMR)

Adaptive mesh refinement allows significant computational savings, especially in the case of complex turbulent 3D flows: highly-resolved flows with large density ratios in complex geometries using unstructured meshes are now accessible. The results obtained in the low-pressure compound nozzle case demonstrate the potential of the method to access a deep numerical insight of jet instabilities and internal flow dynamics with 3D adaptive unstructured meshes. This is a significant progress for the computation of realistic industrial flows.

The refinement strategy can be extended and feature more options, such as combining the present distance criterion with a physical-based one. For instance, a vorticity-based criterion as in [205]:

$$\frac{|\boldsymbol{\nabla} \times \mathbf{u}|\Delta x}{\max\left(|\mathbf{u}|\right)} < \varepsilon_\omega$$

with ε_ω a user-defined threshold parameter. Indeed, the regions with high interface curvature are often associated with high vorticity values [205]. Grid refinement directly based on interface curvature resolution as in the works of [21, 22] seems very challenging with our framework. As mentioned in Chapter 2, the proposed interface-capturing procedure is neither designed for the interface to see various cell sizes at its close vicinity, nor to interpolate at the interface itself.

A strong limitation of the current adaptation methodology is the imposition of a mesh size which needs to be homogeneous in all directions, so that remeshing is achieved with quasi-isotropic tetrahedral elements. However, the physical phenomena to describe, such as boundary or shear layers, are in most cases occuring with a preferential direction. As a consequence, an anisotropic AMR technique can be used to adapt the grid using anisotropic tetrahedral cells, in accordance with the preferential direction of the flow of interest. This strategy, currently in development in YALES2 [206], would lead to substantial computational gains. Some of the resulting challenges are the preservation of the accuracy, stability, and robustness of the numerical schemes, as well as the definition of meaningful physics-based refinement criteria. For interfacial liquid-gas flows, a central challenge lies in the computation of the interface curvature. A comparison between a triangular isotropic grid and an anisotropic one is illustrated in Fig. 6.13 [207]:

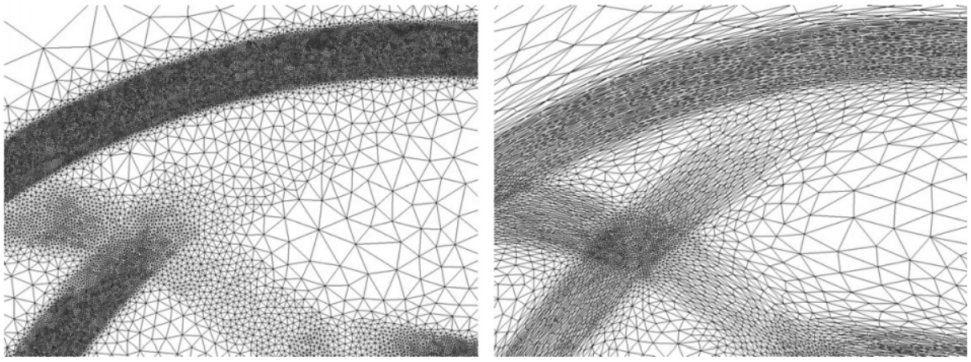


Figure 6.13: Isotropic triangular mesh (Left) versus Anisotropic triangular mesh (Right). Reprinted from [207].

Euler-Lagrange multi-scale strategy

To further increase the computational gain, especially in the liquid jet in crossflow simulations where many droplets need to be resolved in the domain and to be transported far from the injection location, an Eulerian-Lagrangian strategy designed for tetrahedral grids has been proposed to treat the small and spherical droplets as Lagrangian Point Particles (LPP), allowing a strong local coarsening of the tetrahedral mesh, as displayed in Fig. 6.14. Classical Euler-to-Lagrange conversion criteria based on volume, aspect ratio, and distance of the candidate droplets from the resolved interface, found in the literature for cartesian grids, have been adapted to unstructured meshes. The multi-scale coupling results in lighter simulations in terms of mesh size at equivalent interface resolutions.

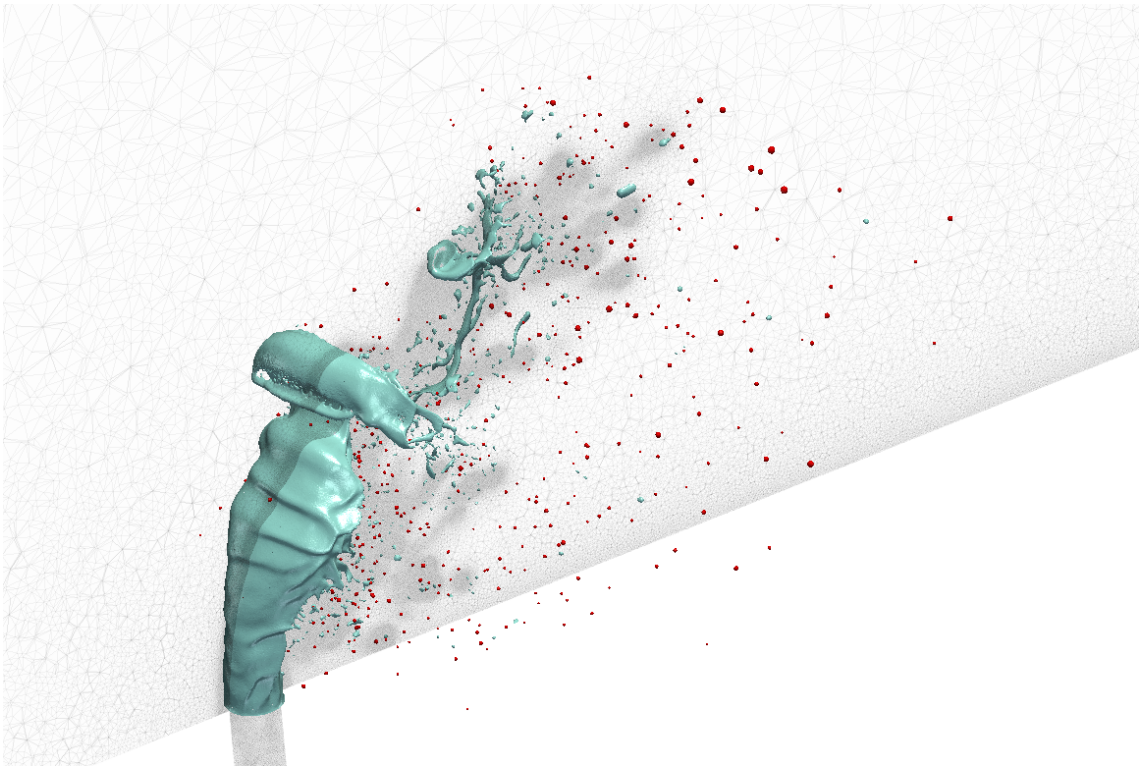


Figure 6.14: Liquid kerosene jet in crossflow using the multi-scale Euler-Lagrange strategy.

Further validations are required to validate the physics, such as comparing converged droplet statistics for the Eulerian-Lagrangian jet in crossflow case versus the fully Eulerian one, and versus experiments. To increase the level of sophistication of the direct conversion procedure (Euler to Lagrange), more physical conversion criteria could be added, such as the internal energy of a droplet to detect whether the drop of interest oscillates due to surface tension or not. After the present work, more focused on the direct conversion algorithm with forward coupling and strong local coarsening of the grid, a robust and accurate two-way coupling strategy can now be integrated in the solver. To this aim, the volume-filtered formulation of the momentum transfer term of Capecelatro and Desjardins (2013) [197] could be used, with the improvements of Evrard et al. (2020) [198] to account for the velocity disturbance of the particle on the flow, for the procedure to be still accurate when the Lagrangian particle is not much smaller than the local cell size. Eventually, the reverse conversion (Lagrange to Euler) can be implemented to account for the impact of an LPP onto the resolved interface. In this case, strong refinement of the mesh is

necessary and constitutes a challenge to address in the future.

Secondary atomization and spray-wall interactions

This thesis has dealt with both interface-resolved primary atomization and Lagrangian modeling of the resulting droplets. To better numerically predict the outcome of an atomizing spray in terms of granulometry (droplet size distribution, SMD), secondary atomization models, such as the one of Gorokhovski (2001) [208, 181], can be added to the presented Lagrangian modeling framework. Moreover, a film model can also be used to improve the numerical prediction of spray-wall interactions, and thus account for the droplet size distribution that comes from film breakup [209]: this is of first importance for airblast atomization simulations.

Towards multi-scale vaporizing liquid-gas flows

Furthermore, this work paves the way to both interface-resolved and Lagrangian evaporation. For Eulerian simulations of vaporizing two-phase liquid-gas flows, the Navier-Stokes equations presented in Chapter 1 together with jump conditions need to be extended and thus new challenges emerge [210]. In addition to the mass and momentum equations (Eqs. (1.1) and (1.2), respectively), the conservation equation of species in the gas phase (in the case of pure liquid) also needs to be considered [211]:

$$\frac{\partial(\rho Y)}{\partial t} + \boldsymbol{\nabla} \cdot (\rho Y \mathbf{u}) = \boldsymbol{\nabla} \cdot (\rho D \boldsymbol{\nabla} Y) \tag{6.3}$$

with Y and D the vapor mass fraction and diffusivity, respectively. For non-reacting problems, the remaining inert gas is $1 - Y$. The following energy conservation equation also supplies the incompressible equation system in case of phase change [211]:

$$\frac{\partial(\rho C_p T)}{\partial t} + \boldsymbol{\nabla} \cdot (\rho C_p T \mathbf{u}) = \boldsymbol{\nabla} \cdot (\lambda \boldsymbol{\nabla} T) \tag{6.4}$$

where T is the fluid temperature, C_p is the specific heat under constant pressure, and λ is the thermal conductivity. The additional physics related to phase change induce more complex interfacial jump conditions. The velocity is not continuous due to mass transfer across the interface Γ, and hence the normal kinematic jump condition writes:

$$[\mathbf{u} \cdot \mathbf{n}]_\Gamma = \dot{m} \left[\frac{1}{\rho}\right]_\Gamma \tag{6.5}$$

with $\dot{m}$ the vaporization rate. This key quantity for phase change also appears in the normal momentum jump condition:

$$[p]_\Gamma = p_{l,\Gamma} - p_{g,\Gamma} = \sigma\kappa + 2\,[\mu]_\Gamma\, \mathbf{n}^T \cdot \boldsymbol{\nabla}\mathbf{u} \cdot \mathbf{n} - \dot{m}^2 \left[\frac{1}{\rho}\right]_\Gamma \tag{6.6}$$

In addition, other discontinuities, such as a heat flux jump, appear implicitly in the definition of $\dot{m}$. Furthermore, as the continuity equation does not reduce to $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$, the transport equation of the smooth phase indicator ψ (if ACLS is employed) now writes:

$$\frac{\partial \psi}{\partial t} + \mathbf{u} \cdot \boldsymbol{\nabla}\psi = 0 \tag{6.7}$$

To summarize the main numerical challenges emerging when interface-resolved phase change is considered [212]:

(i) The computation of the evaporation rate $\dot{m}$, which appears in all interfacial jump conditions. This implies the computation of additional quantities at the interface Γ to handle the new discontinuities, such as temperature T_Γ and vapor mass fraction Y_Γ, and a thermodynamic closure;

(ii) The velocity discontinuity $[\mathbf{u}\cdot\mathbf{n}]_\Gamma$, which needs to be accounted for in the projection method and in all transport equations;

(iii) Interface regression caused by phase change;

(iv) Solving of temperature T and vapor mass fraction Y conservation equations.

Regarding Lagrangian evaporation, the equations describing particle motion Eqs. (5.5) and (5.6) are supplied with an equation for the diameter reduction of the particle. The reduction of $d_p(t)$ depends on the particle evaporation rate $\dot{m}_p$ (mass transfer). In order to completely characterize the evaporation process, a particle temperature $T_p(t)$ evolution equation is also required (heat transfer). In the case of multi-component vaporization, the evolution of the mass fraction $Y_{p,k}$ of each specie k of the particle p is also written [213]. In YALES2, the Spalding evaporation model (1953) [214] extended with the Abramzon and Sirignano model (1989) [215] to account for the finite thickness of the thermal and mass boundary layers around the droplet is available.

Long résumé

Les écoulements diphasiques liquide-gaz sont omniprésents dans la nature et dans les systèmes industriels, et la compréhension de la formation des sprays est cruciale pour concevoir des technologies d'injection complexes. Plus précisément, l'atomisation turbulente est de première importance dans les chambres de combustion aéronautiques, dans lesquelles le carburant est injecté sous forme liquide, puis subit les processus consécutifs d'atomisation, d'évaporation, de mélange turbulent et enfin de combustion.

Tous ces processus physiques sont fortement interdépendants et, par conséquent, l'amélioration de la prédiction numérique de la formation de polluants nécessite des connaissances numériques approfondies du processus d'atomisation. Celui-ci désigne la fragmentation de grandes structures liquides homogènes en petites gouttelettes, et gouverne ainsi la distribution des gouttelettes de carburant dans la chambre de combustion, et par suite l'évaporation. La structure d'un spray s'atomisant peut être décomposée en deux régions principales : les régimes d'atomisation primaire et secondaire.

Afin de fidèlement simuler un spray en atomisation, l'évolution temporelle de la topologie de l'interface liquide-gaz ainsi que les effets de tension de surface doivent être représentés avec précision sur une grille numérique. Pour cela, des techniques de suivi d'interface sont classiquement employées, telles que les méthodes Volume Of Fluid (VOF) [3] ou Level Set (LS) [4, 5], et sont couplées à un modèle numérique de tension de surface. Les géométries réalistes d'injecteurs industriels pour les simulations d'atomisation primaire sont très complexes, incluant notamment des pointes et des coins étroits, ce qui complique l'étape de maillage avec des grilles cartésiennes : l'utilisation de maillages non structurés est donc bien adaptée à ce type de simulations [6]. Bien que la littérature demeure limitée sur les calculs d'atomisation en non structuré, des études antérieures sur des maillages statiques peuvent être trouvées, telles que [7, 8, 9, 10, 11, 12, 13, 14] pour les techniques de capture d'interface Volume Of Fluid (VOF), ou [15, 16, 17, 18, 19] pour les méthodes Level Set (LS).

En dépit de nombreuses années de recherche, les simulations haute performance et haute fidélité de l'atomisation demeurent un défi : la large gamme d'échelles temporelles et spatiales impliquées induit d'importants coûts de calcul. Afin de pallier à ce problème, l'utilisation de techniques d'adaptation de maillage dynamique (AMR) est particulièrement utile, permettant l'allocation des ressources de calcul à la dynamique de l'interface. Une technique populaire pour simuler les écoulements diphasiques incompressibles sur des grilles cartésiennes adaptatives est l'AMR avec structure de données arborescente (quad/octree), comme dans les travaux de Popinet [20, 21]. Concernant les maillages non structurés, Dai et Schmidt [22] ont proposé un algorithme de maillage mobile pour simuler des écoulements à surface libre avec de grandes déformations sur des grilles tétraédriques adaptatives. Les deux études sont proposées dans le cadre des méthodes VOF, et les maillages sont généralement raffinés avec un critère de courbure locale de l'interface. Pour les méthodes classiques Level Set sur des grilles tétraédriques adaptatives, Morgan et Waltz [23] ont introduit une méthode pour traiter les fronts évolutifs avec vitesse prescrite, tandis que Rodriguez et al. ont présenté un solveur CFD parallèle à deux phases utilisant une formulation par éléments finis dans [24], où le maillage adaptatif est raffiné en fonction de la distance à l'interface.

Cependant, même avec l'aide de techniques AMR, les écoulements interfaciaux liquide-gaz d'intérêt pratique peuvent encore présenter des coûts de calcul prohibitifs, en raison du nombre élevé de gouttelettes formées pendant le processus d'atomisation, qui doivent être résolues dans le

domaine numérique en utilisant la grille adaptative. Ainsi, pour augmenter davantage l'efficacité et la précision des calculs de tels écoulements, en particulier la phase dispersée où seules des gouttes isolées sont présentes, des techniques hybrides Eulériennes-Lagrangiennes sont apparues au cours de la dernière décennie, telles que [25, 26, 27, 28, 1], qui emploient des grilles cartésiennes. L'idée est de convertir les petites gouttes sphériques Eulériennes résolues en particules Lagrangiennes ponctuelles (LPP). Avec une telle stratégie de modélisation multi-échelles, les gouttes converties, généralement mal résolues par la grille, sont traitées plus efficacement et plus précisément, et n'ont plus besoin d'être résolues : le déraffinement local du maillage peut ainsi être effectué dynamiquement avec utilisation de l'AMR [26, 28]. Par conséquent, un brouillard de gouttes peut être transporté beaucoup plus loin dans un domaine numérique sans utiliser un grand nombre de cellules, qui seraient nécessaires pour transporter de nombreuses gouttes résolues sur une grande distance.

Dans ce travail, la méthode ACLS (Accurate Conservative Level Set) est utilisée pour suivre l'interface sur des maillages 2D et 3D non structurés (triangulaires et tétraédriques) : cette méthode prédit avec précision la dynamique de l'interface tout en conservant un indicateur de phase lissé. La méthode ACLS n'est donc pas strictement conservative en masse, mais les erreurs sur la conservation de la masse liquide demeurent faibles et maîtrisées [29, 30, 31]. La méthode ACLS repose sur l'utilisation d'une fonction tangente hyperbolique pour représenter l'interface, comme dans le schéma THINC de Xiao et al. [32, 33], proposé précédemment dans le contexte des méthodes VOF algébriques. La fonction tangente hyperbolique ACLS est advectée par l'écoulement fluide, puis remodelée en tangente hyperbolique à l'aide d'une équation de réinitialisation. La méthode ACLS sur grilles non structurées a été proposée précédemment dans [16, 17] sur maillages statiques, et par Antepara et al. [34] pour des grilles tétraédriques adaptatives. Dans ce dernier cas, un AMR non conforme avec une structure de données octree est utilisé. Dans le présent travail, des grilles non structurées adaptatives massivement distribuées sont employées, et adaptées grâce à la librairie de remaillage MMG de manière isotrope, en utilisant à la fois des critères de qualité basés sur la métrique et sur l'allongement des cellules. Nous étendons ces précédentes études ACLS non structurées en incluant une méthode de réinitialisation adaptée et améliorée, initialement introduite par Chiodi et Desjardins sur grilles cartésiennes [35], et qui n'a jamais été implémentée pour des grilles non structurées préalablement à ce travail. En particulier, les propriétés de fusion de plusieurs interfaces de cette dernière équation de réinitialisation sont discutées. En effet, l'équation ACLS originale implique une estimation numérique du gradient de tangente hyperbolique, qui est difficile à calculer avec précision, particulièrement sur maillages non structurés. Elle est donc susceptible d'induire une déformation parasite de l'interface. L'équation reformulée, quant à elle, tire parti d'une fonction level set distance classique tout en préservant beaucoup mieux la forme de l'interface. De plus, dans le cadre proposé, une bande étroite autour de l'interface liquide-gaz est construite afin de calculer les caractéristiques de l'interface uniquement dans la zone d'intérêt [36], permettant des économies de calcul importantes. Par ailleurs, afin de calculer les normales et la courbure de l'interface, une fonction distance signée est reconstruite en parallèle aux nœuds de la bande étroite à l'aide d'une méthode de projection géométrique à marqueurs multiples (GPMM). Cette méthode repose sur une triangulation de l'interface et une projection géométrique exacte sur les éléments de surface les plus proches des noeuds considérés. La tension de surface est traitée explicitement à l'aide de la méthode Ghost-Fluid (GFM) [37]. La procédure de suivi d'interface sur grilles adaptatives non structurées est ensuite couplée à une stratégie de modélisation Lagrangienne, conçue pour des maillages tétraédriques : les petites gouttes résolues et sphériques sont converties en particules Lagrangiennes ponctuelles (LPP), en utilisant des critères de conversion basés sur des considérations numériques et géométriques. La transformation inverse, à savoir la

conversion Lagrangienne-Eulérienne, nécessite la mise en œuvre d'un algorithme de ré-impact d'une particule Lagrangienne sur l'interface résolue. Un tel algorithme n'est pas encore implémenté dans la présente procédure multi-échelle. En conséquence, seules les gouttelettes situées à une certaine distance de l'interface résolue seront converties. Une fois qu'une gouttelette Eulérienne résolue est transformée, le maillage tétraédrique est déraffiné automatiquement et localement autour de la particule lagrangienne nouvellement introduite.

Ce manuscrit est divisé en deux grandes parties. La première partie se concentre sur le solveur Eulérien. Après un premier chapitre général sur la modélisation des écoulements liquide-gaz diphasiques incompressibles, les différentes composantes de la procédure implémentée dans le code non structuré incompressible massivement parallèle YALES2 [38] sont présentées, à savoir la technique ACLS pour le suivi de l'interface, l'intégration à pas fractionnaire des équations de Navier-Stokes avec traitement explicite des conditions de saut, et la stratégie d'adaptation dynamique de maillage. La précision, la convergence spatiale et l'efficacité de calcul de la méthode ACLS non structurée sont évaluées sur des cas test de transport d'interface classiques. Le comportement de l'algorithme global dans le cas d'écoulements diphasiques capillaires canoniques est ensuite vérifié. Enfin, deux simulations LES avec de grands rapports de densité en géométrie complexe sont réalisées avec des maillages tétraédriques adaptatifs : la rupture d'un jet liquide turbulent issu d'un injecteur composé basse pression [39, 40] et d'un jet liquide haute pression non réactif en écoulement transverse sont présentés à différentes résolutions, démontrant l'accessibilité des instabilités de jet avec des maillages non structurés adaptatifs 3D. Les résultats sont confrontés à des expériences réalisées dans les mêmes conditions [41, 42]. La deuxième partie de ce manuscrit présente la stratégie de modélisation multi-échelle hybride Eulérienne-Lagrangienne, conçue pour des grilles tétraédriques adaptatives. Les équations résolues par le solveur Lagrangien sont d'abord dérivées et les hypothèses sous-jacentes explicitement mentionnées. Ensuite, la procédure de conversion entre les deux points de vue et son couplage avec l'adaptation dynamique de maillage (AMR) sont détaillés. La stratégie hybride est finalement appliquée et testée sur des cas canoniques, et sur le même jet liquide haute pression en écoulement transverse que celui présenté en première partie de ce travail. Il permet ainsi d'évaluer les performances de calcul de la procédure de couplage multi-échelle proposée, en utilisant des grilles non structurées adaptatives massivement distribuées.

Le solveur Eulérien diphasique liquide-gaz de YALES2 est présenté en détails dans un premier temps. Un bref aperçu du cadre mathématique et la procédure complète sont d'abord donnés. Ensuite, l'implémentation numérique détaillée de l'algorithme de suivi d'interface ACLS (Accurate Conservative Level Set) pour maillages non structurés (triangulaires et tétraédriques) est présentée. En particulier, les équations sont discrétisées à l'aide de schémas Volumes Finis centrés. L'accent est mis sur la reconstruction géométrique de la fonction distance signée à l'interface pour le calcul de la normale et de la courbure, et sur la discrétisation de l'équation de réinitialisation. Le couplage cohérent du transport d'interface avec le solveur incompressible, nécessaire afin de limiter les erreurs numériques sur la quantité de mouvement, est ensuite détaillé, et les forces de tension de surface, traitées à l'aide de la méthode Ghost-Fluid (GFM), sont intégrées dans la méthode de projection à travers le saut de pression à l'interface, lui-même inclus dans le gradient de pression. L'intégration en temps étant explicite, ces forces capillaires contraignent le pas de temps, afin de pouvoir résoudre les ondes capillaires les plus rapides du domaine. Enfin, la stratégie isotrope d'adaptation dynamique de maillage (AMR), utilisée pour concentrer les mailles autour de l'interface liquide-gaz, est présentée. Une attention particulière est portée sur le contrôle du gradient de taille de maille en dehors de la zone la plus fine.

Une fois la procédure complète présentée, il nous faut valider les différentes composantes du

solveur diphasique de la plateforme YALES2, en utilisant des maillages non structurés statiques et adaptatifs. La précision et la convergence de la reconstruction de la distance signée (algorithme GPMM) et du calcul de la courbure de l'interface, une question centrale pour la simulation des écoulements interfaciaux, sont d'abord évaluées à la fois en 2D et en 3D pour des interfaces circulaires et sphériques, respectivement. Cette étude nous apprend notamment que l'algorithme GPMM pour la reconstruction géométrique de la fonction distance signée converge en moyenne à l'ordre 2 en espace (norme L_2 en 2D et en 3D), et que la courbure calculée converge à l'ordre 0.5 en moyenne (norme L_2 en 2D et en 3D) lorsqu'un opérateur Hessienne non compact est utilisé. Puis, des cas tests classiques de transport d'interface, comme le serpentin ou le disque de Zalesak, sont effectués avec des niveaux d'erreur quantifiés et une convergence spatiale évaluée entre les ordres 1 et 2 pour l'algorithme de suivi d'interface. Enfin, la capacité du solveur à simuler correctement les écoulements pilotés par la tension superficielle et à prédire implicitement les changements de topologie (rupture et fusion d'interfaces ; ce qui n'est pas nécessairement garanti en discrétisant la nouvelle équation de réinitialisation), est vérifiée par la simulation d'écoulements diphasiques canoniques, comme l'amortissement d'une onde capillaire 2D ou l'instabilité de Rayleigh-Plateau. Ces cas démontrent la capacité du solveur à reproduire fidèlement la physique sur maillages non-structurés, et valident ainsi le calcul de la courbure, elle-même impliquée dans le calcul des forces capillaires. Une simulation de collisions de gouttes est également effectuée, afin de vérifier la capacité à prédire implicitement les changements de topologie avec la nouvelle réinitialisation, couplée à l'adaptation dynamique de maillage.

Le dernier chapitre de la première partie vise à évaluer les capacités du solveur diphasique amélioré de YALES2 à effectuer des simulations LES haute fidélité, massivement parallèles, d'atomisation de jets liquides turbulents complexes, en utilisant des grilles tétraédriques adaptatives. Deux situations physiques différentes sont étudiées, toutes deux avec des rapports de densité élevés : (i) Un jet d'eau basse pression issu d'un injecteur à trois disques dans de l'air au repos ; (ii) Un jet de kérosène haute pression atomisé par un fort écoulement d'air transverse. Alors que l'atomisation dans le cas (i) est principalement régie par les instabilités capillaires et la turbulence, le cas (ii) implique une atomisation assistée par de l'air, et donc pilotée par les effets d'inertie aérodynamique. Les coûts de calcul et les performances des simulations sont analysés, de même que les comportements globaux des jets (angle de spray, trajectoire), validés par des expériences réalisées dans les mêmes conditions physiques.

Ainsi, la première partie de ce manuscrit a soulevé la nécessité d'alléger plusieurs contraintes de calcul pour les simulations d'atomisation, en particulier pour le jet atomisé en écoulement transverse. Premièrement, le grand nombre de gouttelettes produites par le processus d'atomisation augmente considérablement la taille de la grille, même avec adaptation dynamique de maillage (AMR). Ceci entraîne des coûts de calcul prohibitifs, en particulier lorsqu'il s'agit de validation par rapport à des données d'expérience obtenues loin du lieu d'injection. Ensuite, dans de telles simulations pilotées par l'inertie, les gouttes advectées à grande vitesse peuvent forcer plus souvent le déclenchement de l'AMR, entraînant également une augmentation du coût de calcul. Enfin, les plus petites gouttes dans le domaine sont mal résolues en utilisant les techniques classiques de capture d'interface, même avec adaptation dynamique de maillage, et des pertes de masse (même raisonnables) se produisent en utilisant la méthode ACLS. La seconde partie du manuscrit est donc une tentative de résolution de ces problèmes, toujours à l'aide de grilles adaptatives non structurées. L'objectif est de coupler la méthodologie Eulérienne adaptative, présentée dans la première partie de cette thèse, à une technique de modélisation Lagrangienne pour les petites gouttes sphériques proches de la limite de résolution. Les petites gouttelettes Eulériennes (en terme de résolution effective), sphériques, doivent être converties en particules ponctuelles de

manière dynamique, en cours de simulation et, par conséquent, des critères de conversion robustes sont nécessaires pour la mise en place d'une telle technique hybride Euler-Lagrange. L'utilisation de l'adaptation dynamique de maillage ouvre la voie à une stratégie multi-échelle robuste et efficace, permettant de cibler les conditions optimales pour les deux points de vue : gouttes Eulériennes bien résolues, et fort déraffinement local du maillage autour des particules nouvellement introduites pour répondre aux hypothèses lagrangiennes, et ainsi alléger automatiquement la grille.

Afin de respecter la condition de sphéricité lors de la transformation d'une gouttelette Eulérienne en une particule Lagrangienne ponctuelle (LPP), la gouttelette d'intérêt doit être suffisamment petite pour que la tension de surface soit la force prépondérante et maintienne ainsi la forme sphérique de la goutte. Dans ce but, par exemple, un critère basé sur le nombre critique de Weber $\mathrm{We}_{\mathrm{crit}}$, au delà duquel les gouttelettes commencent à se fragmenter, peut être employé [1]. Néanmoins, l'élaboration de critères génériques basés sur la physique est un défi et est laissée pour de futures améliorations de la méthodologie de conversion proposée. Dans ce travail, comme répandu dans la littérature pour les maillages cartésiens, nous proposons d'utiliser des critères principalement basés sur des considérations numériques et géométriques, adaptés ici aux maillages non structurés. Ainsi, les critères de transfert du formalisme Eulérien vers le point de vue Lagrangien retenus sont basés sur: (i) deux volumes de coupure pour les petites et moyennes inclusions liquides (calculés à partir de la résolution effective maximale souhaitée pour convertir); (ii) la forme afin de respecter la condition de sphéricité et ainsi éviter de convertir des ligaments à faible volume; (iii) la distance entre la goutte candidate et l'interface résolue, afin d'éviter une situation de ré-impact, non prise en compte dans ce travail.

Nous pouvons à présent tester et valider la procédure hybride multi-échelle Euler-Lagrange présentée, sur des cas tests canoniques (accélération d'une goutte par un écoulement d'air axial, goutte en chute libre) utilisant des maillages tétraédriques adaptatifs. La stratégie Euler-Lagrange est ensuite appliquée au même jet liquide atomisé par un écoulement d'air transverse que celui mené dans la première partie du manuscrit (Chapitre 4). De ce fait, les topologies de jet et les performances de calcul des simulations hybrides et entièrement Eulériennes peuvent être comparées. En particulier, un gain sur la taille du maillage non structuré (nombre de cellules) est mis en évidence.

En conclusion, une procédure complète, efficace, massivement parallèle et robuste pour le calcul d'écoulements diphasiques liquide-gaz sur des maillages non structurés a été présentée. La convergence spatiale et la précision de la technique de suivi d'interface ACLS, qui comprend la première version non structurée de l'équation de réinitialisation ACLS reformulée, ont d'abord été démontrées par des tests de transport d'interface bien connus, à la fois en 2D et en 3D, et confirmées par les résultats obtenus sur des cas tests canoniques d'écoulements diphasiques liquide-gaz. Les écoulements capillaires avec interface sont correctement calculés sur des grilles triangulaires et tétraédriques, malgré les défis que pose l'utilisation de ces maillages irréguliers. Des jets d'atomisation turbulents 3D complexes ont été simulés, comme le jet issu d'un injecteur triple disque, avec une bonne prédiction de leur dynamique globale comparativement aux expériences.

La précision et la convergence du cadre numérique présenté dans ce manuscrit peuvent être encore améliorées en utilisant les schémas volumes finis d'ordre supérieur de [83], qui se sont avérés améliorer la discrétisation des opérateurs sur des grilles non structurées, et donc le calcul de la courbure de l'interface. Le traitement du terme visqueux dans l'équation de quantité de mouvement, qui n'est pas encore écrit sous forme conservative, et notamment à l'interface peut également être amélioré, afin de pouvoir effectuer des simulations plus précises lorsque la viscosité joue un rôle central, par exemple des calculs à faible nombre de Reynolds ou de Laplace (bulle montante, goutte en chute libre). De plus, des études dédiées au caractère *well-balanced* et de

conservation de la quantité de mouvement doivent être menées (par exemple, évolution temporelle de courants parasites basée sur une échelle de temps dissipative pour des gouttes statiques et en translation [52], advection d'une goutte très dense [203], conservation de l'énergie cinétique d'une goutte oscillante [203]). Un traitement explicite revu, moins contraignant, voire implicite de la tension de surface permettrait d'alléger la forte contrainte de pas de temps capillaire, même si pour ce dernier traitement la suppression complète de cette contrainte n'est pas garantie [164]. A terme, un modèle LES de sous-maille supplémentaire dédié à l'atomisation primaire pourrait également être intégré [204].

L'adaptation dynamique de maillage offre des gains importants sur le coût de calcul, en particulier dans le cas d'écoulements 3D turbulents complexes : simuler des écoulements diphasiques hautement résolus, avec de grands rapports de densité, en géométrie complexe, et en utilisant des maillages non structurés est désormais accessible. Les résultats obtenus dans le cas de l'injecteur à triple disque démontrent le potentiel de la méthode pour accéder à une compréhension numérique approfondie des instabilités de jet, mais aussi de la dynamique d'écoulement interne à l'injecteur avec des maillages non structurés adaptatifs 3D. Ceci constitue une avancée significative pour le calcul d'écoulements diphasiques industriels réalistes, et donc pour la conception et la discrimination des injecteurs.

La stratégie d'adaptation de maillage peut être étendue et inclure davantage d'options, telles que la combinaison du critère de distance actuel avec un critère basé sur la physique. Par exemple, un critère basé sur la vorticité comme dans [205]. En effet, les régions à forte courbure d'interface sont souvent associées à des valeurs de vorticité élevées [205]. Le raffinement de la grille directement basé sur la résolution de la courbure de l'interface comme dans les travaux de [21, 22] semble très difficile avec notre cadre de travail. Comme mentionné dans le chapitre 2, la procédure de suivi d'interface proposée n'est ni conçue pour que l'interface rencontre différentes tailles de cellules à son voisinage proche, ni pour interpoler à l'interface elle-même.

Une forte limitation de la méthodologie d'adaptation actuelle est l'imposition d'une taille de maille qui doit être homogène dans toutes les directions, de sorte que le remaillage soit réalisé avec des éléments tétraédriques quasi-isotropes. Cependant, les phénomènes physiques à décrire, tels que les couches limites ou les couches de cisaillement, se produisent dans la plupart des cas avec une direction préférentielle. En conséquence, une technique AMR anisotrope peut être utilisée pour adapter le maillage à l'aide de cellules tétraédriques anisotropes, conformément à la direction préférentielle que présente l'écoulement d'intérêt. Cette stratégie, actuellement en développement dans YALES2 [206], permettrait des gains de calcul substantiels. Certains des défis qui en résultent sont la préservation de la précision, de la stabilité et de la robustesse des schémas numériques, ainsi que la définition de critères de raffinement significatifs basés sur la physique. Pour les écoulements interfaciaux liquide-gaz, un défi majeur réside dans le calcul de la courbure de l'interface.

Afin d'augmenter davantage le gain en coût de calcul, en particulier pour le jet liquide atomisé par un écoulement d'air transverse où de nombreuses gouttes doivent être résolues dans le domaine et transportées loin du lieu d'injection, une stratégie Eulérienne-Lagrangienne conçue pour grilles tétraédriques a été proposée, pour traiter les petites gouttes sphériques sous forme de particule lagrangiennes ponctuelles (LPP), permettant un fort déraffinement local du maillage tétraédrique. Les critères de conversion classiques Euler vers Lagrange basés sur le volume, la forme, et la distance des gouttes candidates à l'interface résolue, issus de la littérature pour grilles cartésiennes, ont été étendus aux maillages non structurés. Le couplage multi-échelle se traduit par des simulations plus légères en termes de taille de maillage à des résolutions d'interface équivalentes.

De plus amples validations sont nécessaires pour valider la physique des cas les plus com-

plexes, telles que la comparaison de statistiques de gouttes convergées pour le jet Eulérien-Lagrangien en écoulement transverse par rapport au jet entièrement Eulérien, et par rapport aux expériences. Pour élever le niveau de sophistication de la procédure de conversion directe (Euler vers Lagrange), des critères de conversion plus physiques pourraient être ajoutés, tels que l'énergie interne d'une goutte pour détecter si celle-ci oscille dû à sa tension superficielle ou non. Pour la suite de ce travail, plus axé sur l'algorithme de conversion directe avec couplage *one-way* et fort déraffinement local de la grille, une stratégie de couplage *two-way* robuste et précise peut maintenant être intégrée au solveur. Dans ce but, la formulation filtrée en volume du terme de transfert de quantité de mouvement de Capecelatro et Desjardins (2013) [197] pourrait être utilisée, avec les améliorations d'Evrard et al. (2020) [198] pour tenir compte de la perturbation de vitesse de la particule sur l'écoulement, afin que la procédure reste précise lorsque la particule Lagrangienne n'est pas beaucoup plus petite que la taille de cellule locale. Finalement, la conversion inverse (Lagrange vers Euler) peut être mise en œuvre pour tenir compte de l'impact d'une particule Lagrangienne sur l'interface résolue. Dans ce cas, un fort raffinement du maillage est nécessaire et constitue un défi à relever dans le futur.

Cette thèse a traité à la fois de l'atomisation primaire avec interface résolue et de la modélisation Lagrangienne des gouttes qui en résultent. Pour mieux prédire numériquement le résultat d'un spray en atomisation en termes de granulométrie (distribution de taille de gouttes, SMD), des modèles d'atomisation secondaire, comme celui de Gorokhovski (2001) [208, 181], peuvent être ajoutés au cadre de modélisation Lagrangien présenté. De plus, un modèle de film peut également être utilisé pour améliorer la prédiction numérique des interactions spray-paroi, et ainsi rendre compte de la distribution de la taille des gouttelettes qui proviennent de la rupture du film [209] : ceci est de première importance pour l'atomisation aérodynamiquement assistée (par exemple pour l'atomisation de type *airblast*).

De plus, ce travail ouvre la voie à la fois à l'évaporation avec interface résolue, et à l'évaporation Lagrangienne. Pour les simulations Eulériennes d'écoulements liquide-gaz diphasiques évaporants, les équations de Navier-Stokes présentées dans le chapitre 1 ainsi que les conditions de saut doivent être étendues et ainsi de nouveaux défis émergent [210]. En plus des équations de masse et de quantité de mouvement, l'équation de conservation des espèces en phase gazeuse (dans le cas de liquide pur) doit également être prise en compte [211]. En outre, la physique supplémentaire liée au changement de phase induit des conditions de saut à l'interface plus complexes. La vitesse n'est pas continue en raison du transfert de masse à travers l'interface. Ainsi la condition de saut cinématique normale s'écrit avec $\dot{m}$ le taux de vaporisation. Cette quantité clé pour le changement de phase apparaît également dans la condition normale de saut de quantité de mouvement. De plus, d'autres discontinuités, comme un saut de flux de chaleur, apparaissent implicitement dans la définition de $\dot{m}$. Par ailleurs, comme l'équation de continuité ne se réduit pas à $\boldsymbol{\nabla} \cdot \mathbf{u} = 0$, l'équation de transport de l'indicateur de phase lisse ψ doit être modifée en conséquence. Pour résumer les principaux défis numériques qui émergent lorsque le changement de phase avec interface résolue est considéré [212] :

(i) Le calcul du taux d'évaporation $\dot{m}$, qui apparaît dans toutes les conditions de saut à l'interface. Cela implique le calcul de quantités supplémentaires à l'interface pour gérer les nouvelles discontinuités, telles que la température à l'interface T_Γ et la fraction massique de vapeur à l'interface Y_Γ, et une fermeture thermodynamique ;

(ii) La discontinuité de vitesse $[\mathbf{u} \cdot \mathbf{n}]_\Gamma$, qui doit être prise en compte dans la méthode de projection et dans toutes les équations de transport ;

(iii) La régression d'interface causée par le changement de phase ;

(iv) La résolution des équations de conservation de la température T et de la fraction massique de vapeur Y.

En ce qui concerne l'évaporation Lagrangienne, les équations décrivant le mouvement des particules sont complétées par une équation pour la réduction du diamètre de la particule. La réduction de $d_p(t)$ dépend du taux d'évaporation des particules $\dot{m}_p$ (transfert de masse). Afin de caractériser complètement le processus d'évaporation, une équation d'évolution de la température des particules $T_p(t)$ est également requise (transfert de chaleur). En cas d'évaporation multi-composantes, l'évolution de la fraction massique $Y_{p,k}$ de chaque espèce k de la particule p est également écrite [213]. Dans YALES2, le modèle d'évaporation de Spalding (1953) [214] étendu avec le modèle d'Abramzon et Sirignano (1989) [215] pour tenir compte de l'épaisseur finie des couches limites thermique et massique autour la goutte est disponible.

References

[1] F. Evrard, F. Denner, B. van Wachem, A multi-scale approach to simulate atomisation processes, Int. J. Multiphase Flow 119 (2019) 194-216.

[2] R. Castrejón-Garcia, A.S. Galán, J.R. Castrejón-Pita, A.A. Castrejón-Pita, The fractal dimension of an oil spray, Fractals 11 (2003) 155-161.

[3] R. Scardovelli, S. Zaleski, Direct numerical simulation of free-surface and interfacial flows, Annu. Rev. Fluid Mech. 31 (1999) 567-603.

[4] M. Sussman, P. Smereka, S. Osher, A Level Set Approach for Computing Solutions to Incompressible Two-Phase Flow, J. Comput. Phys. 114 (1) (1994) 146-159.

[5] J.A. Sethian, P. Smereka, Level Set Methods for Fluid Interfaces, Annu. Rev. Fluid Mech. 35 (2003) 341-372.

[6] J. Leparoux, R. Mercier, V. Moureau, H. Musaefendic, Primary atomization simulation applied to a jet in crossflow aeronautical injector with dynamic mesh adaptation, in: Proceedings ICLASS, 14th Triennal International Conference on Liquid Atomization and Spray Systems, 2018.

[7] L. Jofre, O. Lehmkuhl, J. Castro, A. Oliva, A 3D Volume-of-Fluid advection method based on cell-vertex velocities for unstructured meshes, Comput. Fluids 94 (2014) 14-19.

[8] L. Jofre, R. Borrell, O. Lehmkuhl, A. Oliva, Parallel load balancing strategy for Volume-of-Fluid methods on 3D unstructured meshes, J. Comput. Phys. 282 (2015) 269-288.

[9] C. Ivey, P. Moin, Accurate interface normal and curvature estimates on three-dimensional unstructured non-convex polyhedral meshes, J. Comput. Phys. 300 (2015) 365-386.

[10] C. Ivey, P. Moin, Conservative and bounded volume-of-fluid advection on unstructured grids, J. Comput. Phys. 350 (2017) 387-419.

[11] F. Evrard, F. Denner, B. van Wachem, Estimation of curvature from volume fractions using parabolic reconstruction on two-dimensional unstructured meshes, J. Comput. Phys. 351 (2017) 271-294.

[12] J.K. Patel, G. Natarajan, A novel consistent and well-balanced algorithm for simulations of multiphase flows on unstructured grids, J. Comput. Phys. 350 (2017) 207-236.

[13] B. Xie, P. Jin, H. Nakayama, S. Liao, F. Xiao, A conservative solver for surface-tension-driven multiphase flows on collocated unstructured grids, J. Comput. Phys. 401 (2020) 109025.

[14] T. Maric, D.B. Kothe, D. Bothe, Unstructured un-split geometrical Volume-of-Fluid methods – A review, J. Comput. Phys. 420 (2020) 109695.

[15] M. Herrmann, A balanced force refined level set grid method for two-phase flows on unstructured flow solver grids, J. Comput. Phys. 227 (2008) 2674-2706.

[16] C.E. Kees, I. Akkerman, M.W. Farthing, Y. Bazilevs, A conservative level set method suitable for variable-order approximations and unstructured meshes, J. Comput. Phys. 230 (2011) 4536-4558.

[17] N. Balcazar, L. Jofre, O. Lehmkuhl, J. Castro, J. Rigola, A finite-volume/level-set method for simulating two-phase flows on unstructured grids, Int. J. Multiph. Flow 64 (2014) 55-72.

[18] A. Bøckmann, M. Vartdal, A gradient augmented level set method for unstructured grids, J. Comput. Phys. 258 (2014) 47-72.

[19] S. Lin, J. Yan, D. Kats, G.J. Wagner, A volume-conserving balanced-force level set method on unstructured meshes using a control volume finite element formulation, J. Comput. Phys. 380 (2019) 119-142.

[20] S. Popinet, Gerris: a tree-based adaptive solver for the incompressible Euler equations in complex geometries, J. Comput. Phys. 190 (2003) 572–600.

[21] S. Popinet, An accurate adaptive solver for surface-tension-driven interfacial flows, J. Comput. Phys. 228 (2009) 5838–5866.

[22] M. Dai, D.P. Schmidt, Adaptive tetrahedral meshing in free-surface flow, J. Comput. Phys. 208 (2005) 228–252.

[23] N.R. Morgan, J.I. Waltz, 3D level set methods for evolving fronts on tetrahedral meshes with adaptive mesh refinement, J. Comput. Phys. 336 (2017) 492–512.

[24] J.M. Rodriguez, O. Sahni, R.T. Lahey Jr., K.E. Jansen, A parallel adaptive mesh method for the numerical simulation of multiphase flows, Comput. Fluids 87 (2013) 115–131.

[25] M. Herrmann, A parallel Eulerian interface tracking/Lagrangian point particle multi-scale coupling procedure, J. Comput. Phys. 229 (2010) 745-759.

[26] G. Tomar, D. Fuster, S. Zaleski, S. Popinet, Multiscale simulations of primary atomization, Comput. Fluids 39 (2010) 1864-1874.

[27] Y. Ling, S. Zaleski, R. Scardovelli, Multiscale simulation of atomization with small droplets represented by a Lagrangian point-particle model, Int. J. Multiphase Flow 76 (2015) 122-143.

[28] D. Zuzio, J.-L. Estivalèzes, B. DiPierro, An improved multiscale Eulerian-Lagrangian method for simulation of atomization process, Comput. Fluids 000 (2017) 1-17.

[29] E. Olsson, G. Kreiss, A conservative level set method for two phase flow, J. Comput. Phys. 210 (2005) 225-246.

[30] E. Olsson, G. Kreiss, S. Zahedi, A conservative level set method for two phase flow II, J. Comput. Phys. 225 (2007) 785-807.

[31] O. Desjardins, V. Moureau, H. Pitsch, An accurate conservative level set/ghost fluid method for simulating turbulent atomization, J. Comput. Phys. 227 (2008) 8395-8416.

[32] F. Xiao, Y. Honma, T. Kono, A simple algebraic interface capturing scheme using hyperbolic tangent function, Int. J. Numer. Meth. Fluids 48 (2005) 1023-1040.

[33] F. Xiao, S. Li, C. Chen, Revisit to the THINC scheme: A simple algebraic VOF algorithm, J. Comput. Phys. 230 (19) (2011) 7086-7092.

[34] O. Antepara, N. Balcazar, A. Oliva, Tetrahedral adaptive mesh refinement for two-phase flows using conservative level-set method, Int. J. Numer. Meth. Fluids (2020) 1-23.

[35] R. Chiodi, O. Desjardins, A reformulation of the conservative level set reinitialization equation for accurate and robust simulation of complex multiphase flows, J. Comput. Phys. 343 (2017) 186-200.

[36] D. Adalsteinsson, J.A. Sethian, A Fast Level Set Method for Propagating Interfaces, J. Comput. Phys. 118 (2) (1995) 269-277.

[37] R. Fedkiw, T. Aslam, B. Merriman, S. Osher, A Non-oscillatory Eulerian Approach to Interfaces in Multimaterial Flows (the Ghost Fluid Method), J. Comput. Phys. 152 (2) (1999) 457-492.

[38] V. Moureau, P. Domingo, L. Vervisch, Design of a massively parallel CFD code for complex geometries, Comptes Rendus Mécanique 339 (2-3) (2011) 141-148.

[39] S. Grout, C. Dumouchel, J. Cousin, H. Nuglisch, Fractal analysis of atomizing liquid flows, Int. J. Multiph. Flow 33 (9) (2007) 1023-1044.

[40] J. Cousin, A. Berlemont, T. Menard, S. Grout, Primary breakup simulation of a liquid jet discharged by a low-pressure compound nozzle, Comput. Fluids 63 (2012) 165-173.

[41] J. Becker, C. Hassa, Breakup and atomization of a kerosene jet in crossflow at elevated pressure, Atomization and Sprays 11 (2002) 49-67.

[42] R. Ragucci, A. Bellofiore, A. Cavaliere, Trajectory and momentum coherence breakdown of a liquid jet in high-density air cross-flow, Atomization and Sprays 17 (1) (2007) 47-70.

[43] S. Candel, Mécanique des fluides - Cours, 1995.

[44] P. de Gennes, F. Brochard-Wyart, D. Quere, Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves, Springer New York, 2013.

[45] R. Defay, I. Prigogine, Surface Tension and Adsorption, Wiley, 1966.

[46] J. Gibbs, The Collected Works of J. Willard Gibbs, Yale University Press, 1948.

[47] A. Marchand, J.H. Weijs, J.H. Snoeijer, B. Andreotti, Why is surface tension a force parallel to the interface?, American Journal of Physics 79 (10) (2011) 999–1008.

[48] F. Evrard, Numerical methods for multi-scale interfacial flows, PhD Thesis, Imperial College London, 2017.

[49] G. Tryggvason, R. Scardovelli, S. Zaleski, Direct numerical simulations of gas-liquid multiphase flows, Cambridge University Press, 2011.

[50] S. Popinet, Numerical Models of Surface Tension, Annu. Rev. Fluid Mech. 50 (2018) 49-75.

[51] S. Popinet, Introduction to numerical methods for interfacial flows, Lecture notes, 2011.

[52] M.O. Abu-Al-Saud, S. Popinet, H.A. Tchelepi, A conservative and well-balanced surface tension model, J. Comput. Phys. 371 (2018) 896-913.

[53] J.M. Delhaye, Jump conditions and entropy sources in two-phase systems. Local instant formulation, Int. J. Multiph. Flow 1 (1974) 395-409.

[54] S.P. Lin, Breakup of Liquid Sheets and Jets, Cambridge University Press, 2010.

[55] T. Young, An Essay on the Cohesion of Fluids, Philosophical Transactions of the Royal Society of London 95 (1805) 65-87.

[56] P. Laplace, Supplément Au Dixième Livre Du Traité de Mécanique Céleste: Sur l'action Capillaire, 1806.

[57] A.J. Chorin, The numerical solution of the Navier-Stokes equations for an incompressible fluid, Bull. Amer. Math. Soc. 73 (1967) 928-931.

[58] M. Malandain, Simulation massivement parallèle des écoulements turbulents à faible nombre de Mach, PhD Thesis, INSA de Rouen, 2013.

[59] M. Malandain, N. Maheu, V. Moureau, Optimization of the deflated Conjugate Gradient algorithm for the solving of elliptic equations on massively parallel machines, J. Comput. Phys. 238 (2013) 32-47.

[60] W. Aniszewski, Large Eddy Simulation of Turbulent Two-Phase Flow, PhD Thesis, Czestochowa University of Technology, 2011.

[61] C. Bailly, G. Comte-Bellot, Turbulence, Springer International Publishing, 2015.

[62] L. Guedot, Développement de méthodes numériques pour la caractérisation des grandes structures tourbillonnaires dans les brûleurs aéronautiques : application aux systèmes d'injection multi-points, PhD Thesis, INSA de Rouen, 2015.

[63] V. Moureau, Simulation aux grandes échelles de l'aérodynamique interne des moteurs à piston, PhD Thesis, Ecole Centrale Paris, 2004.

[64] J. Boussinesq, Essai sur la théorie des eaux courantes, Impr. nationale, 1877.

[65] G. Erlebacher, M.Y. Hussaini, C.G. Speziale, T.A. Zang, Toward the large-eddy simulation of compressible turbulent flows, J. Fluid Mech. 238 (1992) 155-185.

[66] J.S. Smagorinsky, General circulation experiments with the primitive equations: I. the basic experiment, Mon. Weath. Rev. 91(3) (1963) 99-164.

[67] P. Domingo-Alvarez, High-pressure combustion large-eddy simulation for an a priori optical diagnostics validation, PhD Thesis, Normandie Université, 2019.

[68] C. Bogey, C. Bailly, Large eddy simulations of round free jets using explicit filtering with-/without dynamic Smagorinsky model, Int. J. Heat Fluid Flow 27 (2006) 603-610.

[69] J.G.M. Kuerten, B.J. Geurts, A.W. Vreman, M. Germano, Dynamic inverse modeling and its testing in large-eddy simulations of the mixing noise, Phys. Fluids 11(12) (1999) 3778-3785.

[70] M. Germano, U. Piomelli, P. Moin, W.H. Cabot, A dynamic subgrid-scale eddy viscosity model, Phys. Fluids A 3 (7) (1991) 1760-1765.

[71] D.K. Lilly, A proposed modification of the Germano subgrid-scale closure method, Phys. Fluids A 4 (3) (1991) 633-635.

[72] M. Germano, Turbulence: the filtering approach, J. Fluid Mech. 238 (1992) 325-336.

[73] C. Meneveau, Germano identity-based subgrid-scale modeling: a brief survey of variations on a fertile theme, Phys. Fluids 24(121301) (2012) 1-14.

[74] J.H. Ferziger, M. Peric, R.L. Street, Computational Methods for Fluid Dynamics - Fourth Edition, Springer International Publishing, 2019.

[75] D. Fuster, G. Agbaglah, C. Josserand, S. Popinet, S. Zaleski, Numerical simulation of droplets, bubbles and waves: state of the art, Fluid Dyn. Res. 41 (2009) 065001.

[76] A. Bakker, Lectures on Applied Computational Fluid Dynamics, www.bakker.org, Dartmouth College, 2008.

[77] Mmg Platform - Robust, Open-source, and Multidisciplinary Software for Remeshing. https://www.mmgtools.org.

[78] P. Bénard, G. Balarac, V. Moureau, C. Dobrzynski, G. Lartigue, Y. D'Angelo, Mesh adaptation for large-eddy simulations in complex geometries, Int. J. Numer. Meth. Fluids 81 (2016) 719-740.

[79] P. Bénard, G. Lartigue, V. Moureau, R. Mercier, Large-Eddy Simulation of the lean-premixed PRECCINSTA burner with wall heat loss, Proceedings of the Combustion Institute 37(4) (2019) 5233-5243.

[80] G. Balarac, F. Basile, P. Bénard, F. Bordeu, J.-B. Chapelier, L. Cirrottola, G. Caumon, C. Dapogny, P. Frey, A. Froehly, G. Ghigliotti, R. Laraufie, G. Lartigue, C. Legentil, R. Mercier, V. Moureau, C. Nardoni, S. Pertant, M. Zakari, Tetrahedral remeshing in the context of large-scale numerical simulation and High Performance Computing, preprint (2021) hal-03344779.

[81] F.H. Harlow, J.E. Welch, Numerical calculation of time-dependant viscous incompressible flow of fluid with a free surface, Phys. Fluids 8 (1965) 2182-2189.

[82] S. Vantieghem, Numerical simulations of quasi-static magnetohydrodynamics using an unstructured finite volume solver: development and applications, PhD Thesis, Université Libre de Bruxelles, 2011.

[83] M. Bernard, G. Lartigue, G. Balarac, V. Moureau, G. Puigt, A framework to perform high-order deconvolution for finite-volume method on simplicial meshes, Int. J. Numer. Meth. Fluids (2020) 1-33.

[84] M. Kraushaar, Application of the compressible and low-Mach number approaches to Large-Eddy Simulation of turbulent flows in aero-engines, PhD Thesis, Université de Toulouse, 2011.

[85] G. Karypis, V. Kumar, A fast and high quality multilevel scheme for partitioning irregular graphs, SIAM Journal on Scientific Computing 20(1) (1999) 359-392.

[86] S. McKee, M.F. Tomé, V.G. Ferreira, J.A. Cuminato, A. Castelo, F.S. Sousa, N. Mangiavacchi, Review - The MAC method, Comput. Fluids 37 (2008) 907-930.

[87] S.O. Unverdi, G. Tryggvason, A front-tracking method for viscous, incompressible, multi-fluid flows, J. Comput. Phys. 100(1) (1992) 25-37.

[88] S. Popinet, S. Zaleski, A front-tracking algorithm for accurate representation of surface tension, Int. J. Numer. Meth. Fluids 30(6) (1999) 775-793.

[89] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juric, N. Al-Rawahi, W. Tauber, J. Han, S. Nas, Y.-J. Jan, A front-tracking method for the computations of multiphase flow, J. Comput. Phys. 169(2) (2001) 708-759.

[90] C.W. Hirt, B.D. Nichols, Volume of fluid (VOF) method for the dynamics of free boundaries, J. Comput. Phys. 39 (1981) 201-225.

[91] J.A. Sethian, An analysis of flow propagation, PhD Thesis, University of California Berkeley, 1982.

[92] J.A. Sethian, Level Set Methods and Fast Marching Methods, Cambridge University Press, 1999.

[93] D. Anderson, G. McFadden, A. Wheeler, Diffuse-interface methods in fluid mechanics, Annu. Rev. Fluid Mech. 30 (1998) 139-165.

[94] V. Badalassi, H. Ceniceros, S. Banerjee, Computation of multiphase systems with phase field models, J. Comput. Phys. 190 (2003) 371-397.

[95] H. Ding, P.D. Spelt, C. Shu, Diffuse interface model for incompressible two-phase flows with large density ratios, J. Comput. Phys. 226 (2007) 2078-2095.

[96] M. El Ouafa, S. Vincent, V. Le Chenadec, Monolithic Solvers for Incompressible Two-Phase Flows at Large Density and Viscosity Ratios, Fluids (2021) 6-23.

[97] P. Germain, Sur l'application de la méthode des puissances virtuelles en mécanique des milieux continus, Comptes rendus hebdomadaire de l'Académie des Sciences A 274(22) (1972) 1051-1055.

[98] P.E. Casal, H. Gouin, Relation entre l'équation de l'énergie et l'équation du mouvement en théorie de korteweg de la capillarité, Comptes-rendus des séances de l'Académie des sciences. Série 2, Mécanique-physique, chimie, sciences de l'univers, sciences de la terre 300(7) (1985) 231–234.

[99] P. Seppecher, Etude d'une modélisation des zones capillaires fluides: interfaces et lignes de contact, PhD Thesis, Ecole Nationale Supérieure de Techniques Avancées, 1987.

[100] D. Jamet, O. Lebaigue, N. Coutris, J. Delhaye, The second gradient method for the direct numerical simulation of liquid–vapor flows with phase change, J. Comput. Phys. 169(2) (2001) 624-651.

[101] Y. Sato, B. Niceno, A conservative local interface sharpening scheme for the constrained interpolation profile method, Int. J. Numer. Meth. Fluids 70 (2012) 441-467.

[102] S. Mirjalili, S.S. Jain, M.S. Dodd, Interface-capturing methods for two-phase flows: An overview and recent developments, Center for Turbulence Research Annual Research Briefs (2017) 117-135.

[103] J. Mariño-Salguero, M. Schäfer, A Modified Normalized Weighting Factor method for improving the efficiency of the blended high-resolution advection schemes in the context of multiphase flows, Exp. and Comput. Multiph. Flow 3(3) (2021) 208-225.

[104] P.H. Gaskell, A.K.C. Lau, Curvature-compensated convective transport: SMART, a new boundedness-preserving transport algorithm, Int. J. Numer. Meth. Fluids 8 (1988) 617-641.

[105] S. Muzaferija, M. Peric, P. Sames, T. Schelin, A two-fluid Navier-Stokes solver to simulate water entry, in: Proceedings of the 22nd Symposium on Naval Hydrodynamics, 1998.

[106] O. Ubbink, R. Issa, A method for capturing sharp fluid interfaces on arbitrary meshes, J. Comput. Phys. 153 (1999) 26-50.

[107] M. Darwish, F. Moukalled, Convective schemes for capturing interfaces of free-surface flows on unstructured grids, Numer. Heat Tr. B-Fund. 49 (2006) 19-42.

[108] J.A. Heyns, A. Malan, T. Harms, O.F. Oxtoby, Development of a compressive surface capturing formulation for modelling free-surface flow by using the volume-of-fluid approach, Int. J. Numer. Meth. Fluids 71 (2013) 788-804.

[109] D. Zhang, C. Jiang, D. Liang, Z. Chen, Y. Yang, Y. Shi, A refined volume-of-fluid algorithm for capturing sharp fluid interfaces on arbitrary meshes, J. Comput. Phys. 274 (2014) 709-736.

[110] B.P. Leonard, The ULTIMATE conservative difference scheme applied to unsteady one-dimensional advection, Comput. Method Appl. M. 88 (1991) 17-74.

[111] S.G. Rubin, P.K. Khosla, Polynomial interpolation methods for viscous flow calculations, J. Comput. Phys. 24 (1977) 217–244.

[112] B.P. Leonard, S. Mokhtari, Beyond first-order upwinding: The ultra-sharp alternative for non-oscillatory steady-state simulation of convection, Int. J. Numer. Meth. Eng. 30 (1990) 729-766.

[113] M.S. Darwish, F. Moukalled, The normalized weighting factor method: A novel technique for accelerating the convergence of high-resolution convective schemes, Numer. Heat Tr. B-Fund. 30 (1996) 217–237.

[114] T. Chourushi, Computationally inexpensive and revised normalized weighting factor method for segregated solvers, Int. J. Comput. Math. 95 (2018) 1622-1653.

[115] S. li, K. Sugiyama, S. Takeuchi, S. Takagi, Y. Matsumoto, F. Xiao, An interface capturing method with a continuous function: The THINC method with multi-dimensional reconstruction, J. Comput. Phys. 231 (2012) 2328-2358.

[116] S. li, B. Xie, F. Xiao, An interface capturing method with a continuous function: The THINC method on unstructured triangular and tetrahedral meshes, J. Comput. Phys. 259 (2014) 260-269.

[117] B. Xie, S. li, F. Xiao, An efficient and accurate algebraic interface capturing method for unstructured grids in 2 and 3 dimensions: The THINC method with quadratic surface representation, Int. J. Numer. Meth. Fluids 76 (2014) 1025-1042.

[118] B. Xie, F. Xiao, Toward efficient and accurate interface capturing on arbitrary hybrid unstructured grids: The THINC method with quadratic surface representation and Gaussian quadrature, J. Comput. Phys. 349 (2017) 415-440.

[119] L. Qian, Y. Wei, F. Xiao, Coupled THINC and level set method: A conservative interface capturing scheme with high-order surface representations, J. Comput. Phys. 373 (2018) 284-303.

[120] L. Qian, Y. Wei, A coupled THINC/QQ and LS framework for simulating incompressible free-surface flows with surface tension, Comput. Fluids 187 (2019) 12-26.

[121] R. Kumar, L. Cheng, Y. Xiong, B. Xie, R. Abgrall, F. Xiao, THINC scaling method that bridges VOF and level set schemes, J. Comput. Phys. 436 (2021) 110323.

[122] R. Debar, Fundamentals of the KRAKEN code, LLNL Tech. Rep. UCIR-760 (1974).

[123] D.L. Youngs, Time-dependent multi-material flow with large fluid distortion, Numer. Methods Fluid Dyn. 1 (1982) 41-51.

[124] R. Scardovelli, S. Zaleski, Interface reconstruction with least-square fit and split eulerian-lagrangian advection, Int. J. Numer. Meth. Fluids 41 (2003) 251-274.

[125] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, Interface reconstruction with least-squares fit and split advection in three-dimensional cartesian geometry, J. Comput. Phys. 225 (2007) 2301-2319.

[126] J.E. Pilliod, E.G. Puckett, Second-order accurate volume-of-fluid algorithms for tracking material interfaces, J. Comput. Phys. 199 (2004) 465-502.

[127] G.H. Miller, P. Colella, A conservative three-dimensional Eulerian method for coupled solid-fluid shock capturing, J. Comput. Phys. 183 (2002) 26-82.

[128] R. Scardovelli, S. Zaleski, Analytical relations connecting linear interfaces and volume fractions in rectangular grids, J. Comput. Phys. 164 (2000) 228-237.

[129] J. Lopez, J. Hernandez, P. Gomez, F. Faura, VOFTools - A software package of calculation tools for volume of fluid methods using general convex grids, Comput. Phys. Commun. 223 (2017) 45-54.

[130] W.J. Rider, D.B. Kothe, Reconstructing volume tracking, J. Comput. Phys. 141 (1998) 112-152.

[131] E.G. Puckett, A.S. Almgren, J.B. Bell, D.L. Marcus, W.J. Rider, A high-order projection method for tracking fluid interfaces in variable density incompressible flows, J. Comput. Phys. 130 (1997) 269-282.

[132] E. Aulisa, S. Manservisi, R. Scardovelli, S. Zaleski, A geometrical area-preserving Volume-of-Fluid advection method, J. Comput. Phys. 192 (2003) 355-364.

[133] G.D. Weymouth, D. K.-P. Yue, Conservative volume-of-fluid method for free-surface simulations on cartesian grids, J. Comput. Phys. 229 (2010) 2853-2865.

[134] J. López, J. Hernández, P. Gómez, F. Faura, A volume of fluid method based on multidimensional advection and spline interface reconstruction, J. Comput. Phys. 195(2) (2004) 718-742.

[135] J. Hernández, J. López, P. Gómez, C. Zanzi, F. Faura, A new volume of fluid method in three dimensions – Part I: Multidimensional advection method with face-matched flux polyhedra, Int. J. Numer. Meth. Fluids 58(8) (2008) 897-921.

[136] M. Owkes, O. Desjardins, A computational framework for conservative, three-dimensional, unsplit, geometric transport with application to the volume-of-fluid (VOF) method, J. Comput. Phys. 270 (2014) 587-612.

[137] M. Owkes, O. Desjardins, A mass and momentum conserving unsplit semi-Lagrangian framework for simulating multiphase flows, J. Comput. Phys. 332 (2017) 21-46.

[138] D. Peng, B. Merriman, S. Osher, H. Zhao, M. Kang, A PDE-Based Fast Local Level Set Method, J. Comput. Phys. 155 (1999) 410-438.

[139] D.L. Chopp, Computing minimal surfaces via Level Set curvature flow, J. Comput. Phys. 106 (1993) 77-91.

[140] D.L. Chopp, Some improvements of the fast marching method, SIAM J. Sci. Comput. 23 (2001) 230-244.

[141] H. Zhao, A fast sweeping method for Eikonal equations, Math Comput. 74 (2005) 603-627.

[142] F. Gibou, R. Fedkiw, S. Osher, A review of level-set methods and some recent applications, J. Comput. Phys. 353 (2018) 82-109.

[143] J.O. McCaslin, O. Desjardins, A localized re-initialization equation for the conservative level set method, J. Comput. Phys. 262 (2014) 408-426.

[144] T. Waclawczyk, A consistent solution of the re-initialization equation in the conservative level-set method, J. Comput. Phys. 299 (2015) 487-525.

[145] M. Sussman, E.G. Puckett, A coupled level set and volume-of-fluid method for computing 3D and axisymmetric incompressible two-phase flows, J. Comput. Phys. 162 (2000) 301-337.

[146] T. Menard, S. Tanguy, A. Berlemont, Coupling level set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet, Int. J. Multiph. Flow 33 (2007) 510-524.

[147] H.T. Ahn, M. Shashkov, Multi-material interface reconstruction on generalized polyhedral meshes, J. Comput. Phys. 226 (2007) 2096-2132.

[148] V. Dyadechko, M. Shashkov, Reconstruction of multi-material interfaces from moment data, J. Comput. Phys. 227 (2008) 5361-5384.

[149] H.T. Ahn, M. Shashkov, Adaptive moment-of-fluid method, J. Comput. Phys. 228 (2009) 2792-2821.

[150] M. Jemison, E. Loch, M. Sussman, M. Shashkov, M. Arienti, M. Ohta, Y. Wang, A Coupled Level Set-Moment of Fluid Method for Incompressible Two-Phase Flows, Journal of Scientific Computing 54 (2013) 454-491.

[151] G. Li, Y. Lian, Y. Guo, M. Jemison, M. Sussman, T. Helms, M. Arienti, Incompressible multiphase flow and encapsulation simulations using the moment-of-fluid method, Int. J. Numer. Meth. Fluids 79 (2015) 456-490.

[152] A.A. Mukundan, T. Menard, J.C. Brändle de Motta, A. Berlemont, A 3D Moment of Fluid method for simulating complex turbulent multiphase flows, Comput. Fluids 198 (2020) 104364.

[153] A.A. Mukundan, T. Menard, J.C. Brändle de Motta, A. Berlemont, A hybrid moment of fluid-level set framework for simulating primary atomization, J. Comput. Phys. 451 (2022) 110864.

[154] D. Gueyffier, A. Nadim, J. Li, R. Scardovelli, S. Zaleski, Volume of fluid interface tracking with smoothed surface stress methods for three-dimensional flows, J. Comput. Phys. 152 (1999) 423-456.

[155] J.U. Brackbill, D.B. Kothe, C. Zemach, A continuum method for modeling surface tension, J. Comput. Phys. 100 (1992) 335-354.

[156] Y. Renardy, M. Renardy, PROST: a parabolic reconstruction of surface tension for the volume-of-fluid method, J. Comput. Phys. 183 (2002) 400-421.

[157] R. Goldman, Curvature formulas for implicit curves and surfaces, Comput. Aided Geom. Des. 22 (2005) 632-658.

[158] J. Poo, N. Ashgriz, A computational method for determining curvatures, J. Comput. Phys. 84 (1989) 483-491.

[159] M. Sussman, A second order coupled level set and volume-of-fluid method for computing growth and collapse of vapor bubbles, J. Comput. Phys. 187 (2003) 110-136.

[160] S.J. Cummins, M.M. Francois, D.B. Kothe, Estimating curvature from volume fractions, Comput. Struct. 83 (2005) 425-434.

[161] P. Karnakov, S. Litvinov, P. Koumoutsakos, A hybrid particle volume-of-fluid method for curvature estimation in multiphase flows, Int. J. Multiph. Flow 125 (2020) 103209.

[162] H. Lamb, Hydrodynamics - 6th Edition, Cambridge University Press, 1932.

[163] C. Galusinski, P. Vigneaux, On stability condition for bifluid flows with surface tension: application to microfluidics, J. Comput. Phys. 227 (2008) 6140-6164.

[164] F. Denner, B.G. van Wachem, Numerical time-step restrictions as a result of capillary waves, J. Comput. Phys. 285 (2015) 24-40.

[165] R. Janodet, G. Vaudor, G. Lartigue, P. Bénard, V. Moureau, R. Mercier, An unstructured conservative level-set algorithm coupled with dynamic mesh adaptation for the computation of liquid-gas flows, in: Proceedings ILASS-Europe, 29th Conference on Liquid Atomization and Spray Systems, 2019.

[166] O. Desjardins, V. Moureau, Methods for multiphase flows with high density ratio, Center for Turbulence Research Proceedings of the Summer Program (2010) 313-322.

[167] C. Dapogny, P. Frey, Computation of the signed distance function to a discrete contour on adapted triangulation, Calcolo 49 (3) (2010) 1-27.

[168] T. Waclawczyk, Modeling of non-equilibrium effects in intermittency region between two phases, Int. J. Multiph. Flow 134 (2021) 103459.

[169] M. Raessi, H. Pitsch, Consistent mass and momentum transport for simulating incompressible interfacial flows with large density ratios using the level set method, Comput. Fluids 63 (2012) 70-81.

[170] Z. Huang, G. Lin, A.M. Ardekani, Consistent, essentially conservative and balanced-force Phase-Field method to model incompressible two-phase flows, J. Comput. Phys. 406 (2020) 109192.

[171] S. Mirjalili, A. Mani, Consistent, energy-conserving momentum transport for simulations of two-phase flows using the phase field equations, J. Comput. Phys. 426 (2021) 109918.

[172] P. Trontin, S. Vincent, J.L. Estivalezes, J.P. Caltagirone, A subgrid computation of the curvature by a particle/level-set method. Application to a front-tracking/ghost-fluid method for incompressible flows, J. Comput. Phys. 231 (2012) 6990-7010.

[173] S.T. Zalesak, Fully multidimensional flux-corrected transport algorithms for fluids, J. Comput. Phys. 31 (3) (1979) 335-362.

[174] R.J. LeVeque, High-resolution conservative algorithms for advection in incompressible flow, SIAM J. Numer. Anal. 33 (2) (1996) 627-665.

[175] A. Prosperetti, Motion of two superposed viscous fluids, Phys. Fluids 24 (1981) 1217-1223.

[176] T. Abadie, J. Aubin, 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, J. Comput. Phys. 297 (2015) 611-636.

[177] C. Weber, Zum Zerfall eines Flüssigkeitsstrahles, Z. Angew. Math. Mech. 11 (1931) 136-141.

[178] A.M. Sterling, C.A. Sleicher, The instability of capillary jets, J. Fluid Mech. 68(3) (1975) 477-495.

[179] N. Ashgriz, J.Y. Poo, Coalescence and separation in binary collisions of liquid drops, J. Fluid Mech. 221 (1990) 183-204.

[180] C. Guillamon, Modeling of Multipoint Fuel Injection for the Large Eddy Simulation of Aeronautical Combustion Chambers, PhD Thesis, Institut National Polytechnique de Toulouse, 2022.

[181] C. Guillamon, R. Janodet, L. Voivenel, R. Mercier, V. Moureau, Building Lagrangian injectors from resolved primary atomization simulations. Application to jet in crossflow fuel injection, in: Proceedings ICLASS, 15th Triennial International Conference on Liquid Atomization and Spray Systems, 2021.

[182] G.G. Stokes, On the Effect of the Internal Friction of Fluids on the Motion of Pendulums, Transactions of the Cambridge Philosophical Society 9 (1851) 8.

[183] Y. Dufresne, Développement d'un solveur 3D massivement parallèle pour la prédiction d'écoulements granulaires réactifs en géométrie complexe, PhD Thesis, INSA Rouen, 2019.

[184] M.R. Maxey, J.R.J. Riley, Equation of motion for a small rigid sphere in a non uniform flow, Phys. Fluids 26(4) (1983) 883.

[185] R. Gatignol, The Faxén formulas for a rigid particle in an unsteady non-uniform Stokes-flow, Journal de Mécanique théorique et appliquée 2(2) (1983) 143-160.

[186] A.B. Basset, A Treatise on Hydrodynamics, with Numerous Examples, Number 2. Bell and co., Cambridge Deighton, 1888.

[187] V. Armenio, V. Fiorotto, The importance of the forces acting on particles in turbulent flows, Phys. Fluids 13(8) (2001) 2437-2440.

[188] J.G.M. Kuerten, Point-Particle DNS and LES of Particle-Laden Turbulent flow - a state-of-the-art review, Flow, Turbulence and Combustion 97(3) (2016) 689-713.

[189] L. Schiller, A. Naumann, Uber die grundlegenden Berechnungen bei der Schwerkraftaufbereitung, Ver. Deut. Ing 77 (1933) 318-320.

[190] M.R. Maxey, B.K. Patel, E.J. Chang, L.P. Wang, Simulations of dispersed turbulent multiphase flow, Fluid Dyn. Res. 20 (1997) 143-156.

[191] C.T. Crowe, M. Sommerfield, Y. Tsuji, Multiphase Flows with Droplets and Particles, CRC Press, 1998.

[192] J.K. Eaton, Two-way coupled turbulence simulations of gas-particle flows using point-particle tracking, Int. J. Multiphase Flow 35 (2009) 792-800.

[193] Y. Ling, M. Parmar, S. Balachandar, A scaling analysis of added-mass and history forces and their coupling in dispersed multiphase flows, Int. J. Multiphase Flow 57 (2013) 102-114.

[194] C.T. Crowe, M.P. Sharma, D.E. Stock, The Particle-Source-In Cell (PSI-CELL) Model for Gas-Droplet Flows, J. Fluids Eng. 99 (1977) 325-333.

[195] L. Brandt, F. Coletti, Particle-Laden Turbulence: Progress and Perspectives, Annu. Rev. Fluid Mech. 54 (2022) 159-89.

[196] F. Evrard, F. Denner, B. van Wachem, Quantifying the errors of the particle-source-in-cell Euler-Lagrange method, Int. J. Multiphase Flow 135 (2021) 103535.

[197] J. Capecelatro, O. Desjardins, An Euler-Lagrange strategy for simulating particle-laden flows, J. Comput. Phys. 238 (2013) 1-31.

[198] F. Evrard, F. Denner, B. van Wachem, Euler-Lagrange modelling of dilute particle-laden flows with arbitrary particle-size to mesh-spacing ratio, J. Comput. Phys.: X 8 (2020) 100078.

[199] M. Gorokhovski, M. Herrmann, Modeling primary atomization, Annu. Rev. Fluid Mech. 40 (2008) 343-366.

[200] R. Di Felice, A relationship for the wall effect on the settling velocity of a sphere at any flow regime, Int. J. Multiphase Flow 22 (1996) 527-533.

[201] R. Clift, W.H. Gauvin, The motion of particles in turbulent gas streams, Proc. Chemeca 1 (1970) 14-28.

[202] A.H. Lefebvre, V.G. McDonell, Atomization and Sprays (2nd Edition), CRC Press, 2017.

[203] G. Vaudor, T. Ménard, W. Aniszewski, M. Doring, A. Berlemont, A consistent mass and momentum flux computation method for two phase flows. Application to atomization process, Comput. Fluids 152 (2017) 204-216.

[204] J. Shinjo, Recent Advances in Computational Modeling of Primary Atomization of Liquid Fuel Sprays, Energies 11(11) (2018) 2971.

[205] D. Fuster, A. Bague, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, S. Scardovelli, S. Zaleski, Simulation of primary atomization with an octree adaptive mesh refinement and VOF method, Int. J. Multiphase Flow 35(6) (2009) 550-565.

[206] G. Janodet, Etude de l'adaptation anisotrope et dynamique de maillages tétraédriques pour des applications CFD, Masters Thesis, Université Paris-Saclay, 2021.

[207] F. Alauzet, P. Frey, P.L. George, B. Mohammadi, 3D transient fixed point mesh adaptation for time dependent problems. Application to CFD simulations, J. Comput. Phys. 222 (2007) 592-623.

[208] M. Gorokhovski, The stochastic Lagrangian model of drop breakup in the computation of liquid sprays, Atomization and Sprays 11 (2001) 505-519.

[209] G. Chaussonnet, Modeling of liquid film and breakup phenomena in Large-Eddy Simulations of aeroengines fueled by airblast atomizers, PhD Thesis, Institut National Polytechnique de Toulouse, 2014.

[210] Y. Atmani, F. Pecquery, M. Cailler, V. Moureau, G. Lartigue, R. Mercier, R. Janodet, G. Sahut, G. Balarac, Consistent scalar transport with front capturing methods: application to two-phase heat transfer, in: Proceedings ICLASS, 15th Triennial International Conference on Liquid Atomization and Spray Systems, 2021.

[211] J. Palmore Jr, O. Desjardins, A volume of fluid framework for interface-resolved simulations of vaporizing liquid-gas flows, J. Comput. Phys. 399 (2019) 108954.

[212] V. Boniou, On the numerical simulation of evaporating two-phase flows using sharp interface capturing methods, PhD Thesis, Université Paris-Saclay, 2021.

[213] H. Larabi, Vers la modélisation multi-composants des flammes de spray, PhD Thesis, INSA de Rouen, 2019.

[214] D.B. Spalding, The combustion of liquid fuels, in: Symposium (international) on combustion 4 (1953) 847-864.

[215] B. Abramzon, W.A. Sirignano, Droplet vaporisation model for spray combustion calculations, Int. J. Heat and Mass Transfer 32(9) (1989) 1605-1618.

[216] R. Janodet, C. Guillamon, V. Moureau, R. Mercier, G. Lartigue, P. Bénard, T. Ménard, A. Berlemont, A massively parallel accurate conservative level set algorithm for simulating turbulent atomization on adaptive unstructured grids, J. Comput. Phys. 458 (2022) 111075.