

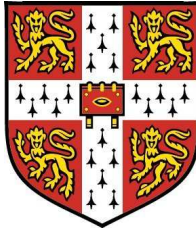
LARGE EDDY SIMULATION OF PRIMARY LIQUID-SHEET BREAKUP

by

Thibault Pringuey

Trinity Hall

University of Cambridge



This dissertation is submitted for the degree of

Doctor of Philosophy

April 2012

*Dedicated to my wife Imelda
and
to my grandfather Maurice.*

Declaration

This dissertation is the result of my own work and includes nothing which is the outcome of work done in collaboration except where specifically indicated in the text. The dissertation contains approximately **63500** words, **98** figures and **22** tables.

Thibault Pringuey

Hopkinson Laboratory, University of Cambridge,
Wednesday, 4th of April, 2012

Acknowledgements

I am very grateful to all people who supported me during this PhD project. First of all, I would like to thank my supervisor, Prof. Stewart Cant, for his invaluable help, his enthusiasm and his constant encouragement. Thanks to his availability, his technical expertise and his great sense of humour this research project has been a very positive and enjoyable experience.

I am also thankful to the senior staff of the Combustion department of Rolls-Royce plc. for funding this work. In particular, I would like to thank Jon Gregory, John Moran and Ken Young who made this project possible. My thanks also go to Marco Zedda and Steve Harding (Rolls-Royce plc.) for their technical input.

I would like to thank Dr. Thomas Rüberg for his useful advice on computational integration. In addition, I thank Prof. Rickard Bensow and Prof. Christer Fureby (Chalmers University of Technology) for welcoming me in Sweden and for our very interesting technical discussions. I am also thankful to Prof. Peter Stephan (Technical University of Darmstadt) and his group for the visit of their experimental facilities and our discussions on the capability of OpenFOAM. The support received from the administrators of Cambridge's High Performance Computing facilities is also gratefully acknowledged.

I would like to express my gratitude to all my colleagues of the laboratory for the pleasant working atmosphere of the Hopkinson Students' Room. In particular, I thank Camille Letty, Chris Bohn, Andrea Maffioli, Teresa Leung, Adam Comer, Ryan Harper, Robert Gordon, Alexandre Neophytou, Andrea Pastore, Giulio Borgh-

esi, Davide Cavaliere and Kieran Hegarty.

My final thanks go to my wife Imelda, for her advice and constant support along the whole project, and to my daughters Claire and Éléonore — born in the course of this PhD — who kept on cheering (day and night).

Abstract

This research project aims at providing the aeronautical industry with a modelling capability to simulate the fuel injection in gas turbine combustion chambers.

The path to this objective started with the review of state-of-the-art numerical techniques to model the primary breakup of liquid fuel into droplets. Based on this and keeping in mind the requirements of the industry, our modelling strategy led to the generation of a mass-conservative method for efficient atomisation modelling on unstructured meshes. This goal has been reached with the creation of high-order numerical schemes for unstructured grids, the development of an efficient numerical method that transports the liquid-vapour interface accurately while conserving mass and the implementation of an algorithm that outputs the droplet boundary conditions to separate combustion codes.

Both high-order linear and WENO schemes have been created for general polyhedral meshes. The notorious complexity of high-order schemes on 3D mixed-element meshes has been handled by the creation of a series of algorithms. These include the tetrahedralisation of the mesh, which allows generality of the approach while remaining efficient and affordable, together with a novel approach to stencil generation and a faster interpolation of the solution. The performance of the scheme has been demonstrated on typical two-dimensional and three-dimensional test cases for both linear and non-linear hyperbolic partial differential equations.

The conservative level set method has been extended to unstructured meshes and its performance has been improved in terms of robustness and accuracy. This was

achieved by solving the equations for the transport of the liquid volume fraction with our novel WENO scheme for polyhedral meshes and by adding a flux-limiter algorithm. The resulting method, named robust conservative level set, conserves mass to machine accuracy and its ability to capture the physics of the atomisation is demonstrated in this thesis.

To be readily applicable to the simulation of atomisation, the novel interface-capturing technique has been embedded in a framework — within the open source CFD code OpenFOAM — that solves the velocity and pressure fields, outputs droplet characteristics and runs in parallel. In particular, the production of droplet boundary conditions involves a set of routines handling the selection of drops in the level set field, the calculation of relevant droplet characteristics and their storage into data files. An n-halo parallelisation method has been implemented in OpenFOAM to perform the computations at the expected order of accuracy.

Finally, the modelling capability has been demonstrated on the simulation of primary liquid-sheet breakup with relevance to fuel injection in aero-engine combustors. The computation has demonstrated the ability of the code to capture the physics accurately and further illustrates the potential of the numerical approach.

Contents

| | |
|---|----------|
| Contents | vii |
| List of figures | xiii |
| List of tables | xvii |
| Nomenclature | xx |
| 1 Introduction | 1 |
| 1.1 The aeronautical application | 2 |
| 1.2 Motivations | 3 |
| 1.3 Approaches to study the primary breakup | 3 |
| 1.4 Numerical modelling of atomisation | 4 |
| 1.4.1 Numerical framework | 4 |
| 1.4.2 Interface description | 5 |
| 1.4.3 Treatment of singularities | 5 |
| 1.5 Aim of the present work | 6 |
| 1.6 Outline of the thesis | 7 |
| 2 Physics of primary breakup | 8 |
| 2.1 Fundamental forces and dimensional analysis | 8 |
| 2.2 Early research interest | 9 |
| 2.3 Round liquid jet breakup | 10 |
| 2.3.1 Jet breakup regimes | 10 |
| 2.3.2 Spray structure | 11 |

| | | |
|----------|--|-----------|
| 2.3.3 | Properties of round jet sprays | 13 |
| 2.3.4 | Ligament formation | 13 |
| 2.4 | Liquid sheet breakup | 16 |
| 2.4.1 | Natural disintegration | 17 |
| 2.4.2 | Assisted non-linear disintegration | 17 |
| 3 | Numerical modelling of multiphase flows | 21 |
| 3.1 | Problem formulation | 22 |
| 3.1.1 | Navier-Stokes equations | 22 |
| 3.1.2 | Incompressible flows | 23 |
| 3.1.3 | Fluid mechanics with interfaces | 23 |
| 3.1.4 | Whole domain formulation | 25 |
| 3.1.5 | Conservative form | 26 |
| 3.2 | Numerical framework | 26 |
| 3.2.1 | Direct numerical simulation of atomisation | 26 |
| 3.2.2 | Large eddy simulation | 32 |
| 3.3 | Interface description | 34 |
| 3.3.1 | Overview | 34 |
| 3.3.2 | Formulation of the VOF method | 41 |
| 3.3.3 | Formulation of the level set method | 48 |
| 3.4 | Treatment of singularities | 53 |
| 3.4.1 | Physical origin of the surface tension | 54 |
| 3.4.2 | Continuum surface force | 55 |
| 3.4.3 | Ghost fluid method | 58 |
| 3.4.4 | Alternative methods | 61 |
| 3.4.5 | Spurious currents | 62 |
| 4 | Multiphase codes | 65 |
| 4.1 | Available codes | 65 |

| | | |
|----------|--|------------|
| 4.1.1 | OpenFOAM | 65 |
| 4.1.2 | Gerris | 66 |
| 4.2 | Code validation | 67 |
| 4.2.1 | Advection algorithm | 67 |
| 4.2.2 | Basic two-phase flows | 77 |
| 4.3 | Simulation of atomisation | 89 |
| 4.3.1 | Simulations of liquid breakup with SGS model | 92 |
| 4.3.2 | Simulations of sheet breakup without SGS model | 101 |
| 5 | Atomisation modelling | 111 |
| 5.1 | Demonstrated numerical capabilities | 111 |
| 5.1.1 | LES with VOF | 111 |
| 5.1.2 | VOF with adaptive mesh refinement | 113 |
| 5.1.3 | Coupled LS-VOF combined with GFM | 116 |
| 5.1.4 | RLSG combined with Lagrangian tracking | 117 |
| 5.1.5 | Conservative LS with GFM | 118 |
| 5.2 | Towards an industry-friendly approach | 119 |
| 5.2.1 | Development of sub-grid scale models | 119 |
| 5.2.2 | Realistic boundary conditions | 120 |
| 5.2.3 | Accurate description of the interface | 121 |
| 5.2.4 | Numerical implementation of the physics | 122 |
| 5.2.5 | Summary | 123 |
| 5.3 | A modelling capability for fuel-injector design | 124 |
| 5.3.1 | Industry requirements | 124 |
| 5.3.2 | Modelling strategy | 125 |
| 5.3.3 | Outline of the solver | 126 |
| 6 | A high-order scheme for general unstructured meshes | 128 |
| 6.1 | WENO schemes for unstructured meshes | 129 |

| | | |
|----------|---|------------|
| 6.2 | Overview of the numerical scheme | 129 |
| 6.3 | Numerical formulation | 132 |
| 6.3.1 | Methodology for the linear reconstruction | 132 |
| 6.3.2 | Methodology for WENO schemes | 145 |
| 6.3.3 | Determination of the numerical flux | 157 |
| 6.4 | Application to the level set equation | 162 |
| 6.4.1 | Finite volume formulation of the level set equation | 162 |
| 6.4.2 | The Riemann problem for the level set equation | 163 |
| 6.5 | Application to the Burgers' equation | 166 |
| 6.5.1 | Finite volume formulation of the Burgers' equation | 166 |
| 6.5.2 | The Riemann problem for the Burgers' equation | 166 |
| 6.6 | Performance of the scheme | 168 |
| 6.6.1 | Level set test cases | 168 |
| 6.6.2 | Numerical convergence study | 176 |
| 6.6.3 | Extension to a non-linear PDE | 181 |
| 7 | Robust conservative level set method | 183 |
| 7.1 | Overview of the method | 184 |
| 7.2 | Transport of the level set | 185 |
| 7.2.1 | Mathematical formulation | 185 |
| 7.2.2 | Finite volume discretisation | 186 |
| 7.2.3 | Temporal discretisation | 191 |
| 7.2.4 | Choice of the parameter ϵ | 193 |
| 7.2.5 | Initialisation of the conservative level set field | 195 |
| 7.3 | Calculation of the interface normal | 196 |
| 7.3.1 | Mathematical formulation | 196 |
| 7.3.2 | Numerical tests | 198 |
| 7.4 | Multidimensional universal limiter with explicit solution | 203 |
| 7.4.1 | Overview of the method | 203 |

| | | |
|----------|---|------------|
| 7.4.2 | Determination of the limiter factor | 205 |
| 7.5 | Performance of the method | 208 |
| 7.6 | Interpretation of the method | 211 |
| 8 | A mass-conservative method for efficient atomisation modelling in parallel | 212 |
| 8.1 | Solution of the incompressible Navier-Stokes equations | 213 |
| 8.1.1 | Conservative formulation | 213 |
| 8.1.2 | Systems of linear algebraic equations | 214 |
| 8.1.3 | Pressure-velocity coupling | 215 |
| 8.2 | Droplet transfer | 221 |
| 8.2.1 | Motivations | 221 |
| 8.2.2 | Outline of the method | 222 |
| 8.2.3 | Identification of blobs | 223 |
| 8.2.4 | Selection of drops | 225 |
| 8.2.5 | Drop characteristics of interest | 229 |
| 8.2.6 | Test cases | 232 |
| 8.3 | Outline of the parallel implementation | 235 |
| 8.4 | Performance of the method on basic two-phase flow problems | 240 |
| 8.4.1 | Rayleigh-Taylor instability | 241 |
| 8.4.2 | Falling drop in a pool | 245 |
| 9 | Simulation of liquid sheet breakup | 251 |
| 9.1 | Quasi-DNS/LES formulation | 252 |
| 9.1.1 | Filtering | 252 |
| 9.1.2 | Filtered Navier-Stokes equations | 254 |
| 9.1.3 | Residual kinetic energy | 255 |
| 9.1.4 | Sub-grid scale modelling | 256 |
| 9.1.5 | Quasi-DNS/LES equations | 259 |

| | | |
|-----------|--|------------|
| 9.2 | Settings of the computation | 260 |
| 9.2.1 | Domain and material properties | 260 |
| 9.2.2 | Choice of RCLS settings | 264 |
| 9.3 | Results and discussion | 270 |
| 9.3.1 | Instabilities of the liquid sheet | 270 |
| 9.3.2 | Torn sheet breakup | 274 |
| 9.3.3 | Breakup length | 277 |
| 10 | Conclusion | 284 |
| 10.1 | Achievements | 284 |
| 10.1.1 | Modelling strategy to simulate fuel injection | 284 |
| 10.1.2 | Novel WENO scheme for unstructured meshes | 286 |
| 10.1.3 | Mass-conservative interface description | 287 |
| 10.1.4 | Modelling capability for the simulation of atomisation | 288 |
| 10.1.5 | Demonstration of the numerical tool on the primary breakup . | 289 |
| 10.2 | Follow-on research topics | 290 |
| 10.2.1 | Application to aeronautical fuel-injectors | 290 |
| 10.2.2 | Improvement of the interface description technique | 290 |
| 10.2.3 | Extension of the modelling capability | 291 |
| 10.2.4 | Development of sub-grid scale models | 292 |
| 10.2.5 | Super-critical fuel injection | 293 |
| | Bibliography | 294 |

List of Figures

| | | |
|-----|---|----|
| 1.1 | Atomisation in a combustion chamber | 2 |
| 2.1 | Plot of the breakup length vs. jet velocity | 10 |
| 2.2 | Structure of a pressure-atomised spray | 12 |
| 2.3 | Photo of the disintegration of a round jet in co-flow | 16 |
| 2.4 | The two modes growing along an inviscid liquid sheet | 17 |
| 2.5 | The two sheet breakup regimes | 19 |
| 3.1 | Main interface description methods | 38 |
| 3.2 | Main interface reconstruction techniques for VOF methods | 46 |
| 3.3 | Interface advection algorithms for VOF methods | 47 |
| 3.4 | Level sets of a water drop falling under gravity | 51 |
| 3.5 | Physical explanation for surface tension | 54 |
| 3.6 | Illustration of the continuum surface force method in 2D | 56 |
| 3.7 | Illustration of the ghost fluid method in 1D | 59 |
| 3.8 | Spurious currents in an equilibrium bubble | 62 |
| 4.1 | Performance of the advection algorithms on the rotation of a 2D cross | 73 |
| 4.2 | Performance of the advection algorithms on the translation of a 2D cross | 74 |
| 4.3 | Performance of the advection algorithms on the slotted disk of Zalesak | 75 |
| 4.4 | Performance of the advection algorithms on a disk in a deformation field | 76 |

| | | |
|------|--|-----|
| 4.5 | Simulation results for the Rayleigh-Taylor instability — OpenFoam vs. Gerris | 82 |
| 4.6 | Performance of OpenFOAM on the Rayleigh-Taylor instability | 83 |
| 4.7 | Performance of Gerris on the Rayleigh-Taylor instability ($\sigma = 0$) | 84 |
| 4.8 | Volume fractions predicted by the codes for the falling drop in a pool — OpenFoam vs. Gerris | 85 |
| 4.9 | Volume fractions predicted by the codes for the phase inversion — Time $t = 1s, 2.25s, 4.5s$ | 86 |
| 4.10 | Volume fractions predicted by the codes for the phase inversion — Time $t = 6.25s, 9.75s, 22.25s$ | 87 |
| 4.11 | Volume fractions predicted by the codes for the phase inversion — Time $t = 34.75s, 42.25s, 100s$ | 88 |
| 4.12 | Simulation of Diesel jet breakup with OpenFOAM — Mesh | 95 |
| 4.13 | Simulation of Diesel jet breakup with OpenFOAM — Onset of the breakup | 96 |
| 4.14 | Simulation of axisymmetric sheet breakup with OpenFOAM — Mesh | 97 |
| 4.15 | Simulation of axisymmetric sheet breakup with OpenFOAM — Onset of the breakup | 98 |
| 4.16 | Simulation of axisymmetric sheet breakup with OpenFOAM — Breakup mechanism A | 99 |
| 4.17 | Simulation of axisymmetric sheet breakup with OpenFOAM — Breakup mechanism B | 100 |
| 4.18 | Simulation of flat sheet breakup with Gerris — Mesh | 105 |
| 4.19 | Simulation of flat sheet breakup with Gerris — Build-up of instability | 106 |
| 4.20 | Simulation of flat sheet breakup with Gerris — Onset of the breakup | 107 |
| 4.21 | Simulation of flat sheet breakup with Gerris — $We = 410$ | 108 |
| 4.22 | Simulation of flat sheet breakup with Gerris — $We = 2100$ | 109 |
| 4.23 | Simulation of flat sheet breakup with Gerris — $We = 9300$ | 110 |

| | | |
|------|---|-----|
| 5.1 | LES-VOF simulation of a Diesel spray atomisation | 112 |
| 5.2 | LES-VOF simulation of a Diesel spray atomisation (fine mesh) | 113 |
| 5.3 | VOF simulation of a swirl outward-opening jet | 115 |
| 5.4 | CLSVOF/GFM simulation of a Diesel spray atomisation | 116 |
| 5.5 | RLSG/Lagrangian tracking simulation of a jet in co-flow | 118 |
| 5.6 | CLS/GFM of Diesel spray atomisation | 119 |
| 6.1 | Point-neighbour approach in 2D | 135 |
| 6.2 | Schematic of the point-neighbours of a targeted cell | 137 |
| 6.3 | Layers of cells added to the dynamic list at each iteration | 138 |
| 6.4 | Schematic of the edge-neighbours of a given vertex in a cell | 139 |
| 6.5 | Tetrahedralisation of a convex polyhedron | 141 |
| 6.6 | Tetrahedral decomposition ensuring convergence | 142 |
| 6.7 | Ten-cell sectoral stencils of E_0 coloured by sector | 150 |
| 6.8 | Mapping to the first octant $(+, +, +)$ | 151 |
| 6.9 | Thirteen-cell sectoral stencils of E_0 coloured by sector — fast search procedure | 153 |
| 6.10 | Rotated Cartesian frame in 2D: (n_l, s_l) | 164 |
| 6.11 | Meshes for the level set test cases | 171 |
| 6.12 | Zero level set for the translation of a slotted disk and the disk in a deformation field | 172 |
| 6.13 | Zero level set for the sphere in a deformation field | 175 |
| 6.14 | L^2 error vs. normalised CPU time for the WENO3 applied to the linear equation | 179 |
| 6.15 | L^2 error vs. normalised CPU time for the WENO4 applied to the linear equation | 180 |
| 6.16 | Solution for the 3D Burgers' equation on the hybrid mesh | 182 |
| 7.1 | Contour plots of the CLS field for gradient performance tests | 198 |

| | | |
|------|--|-----|
| 7.2 | Comparative performance of the gradient calculation: low order vs. high order — Gradient magnitudes | 201 |
| 7.3 | Comparative performance of the gradient calculation: low order vs. high order — Gradient in the direction x | 202 |
| 7.4 | Performance of the transport algorithms of <i>interFoam</i> and the RCLS method | 209 |
| 7.5 | Performance of the transport algorithms of the ACLS and the RCLS methods | 210 |
| 8.1 | Schematic of a blob on a 1D mesh | 225 |
| 8.2 | Illustration of the principle of face-neighbours used to define blobs | 226 |
| 8.3 | Illustration of the direction criterion for the droplet selection | 228 |
| 8.4 | Illustration of the first droplet removal technique: “option 1” | 230 |
| 8.5 | Illustration of the second droplet removal technique: “option 2” | 231 |
| 8.6 | Illustration of the droplet transfer on the static test case | 234 |
| 8.7 | Illustration of the droplet transfer on the Rayleigh-Taylor instability case | 236 |
| 8.8 | Decomposition of the domain using a 0-halo approach | 237 |
| 8.9 | Decomposition of the domain using a n-halo approach | 238 |
| 8.10 | Interface predicted by <i>RCLSFoam</i> for the Rayleigh-Taylor instability — <i>RCLSFoam</i> vs. <i>interFoam</i> | 247 |
| 8.11 | Volume fractions and interface predicted by <i>RCLSFoam</i> for the Rayleigh-Taylor instability — Hybrid vs. triangular mesh | 248 |
| 8.12 | Interface predicted by <i>RCLSFoam</i> for the Rayleigh-Taylor instability — Comparative performance parallel approaches | 249 |
| 8.13 | Volume fractions and interface predicted by <i>RCLSFoam</i> for the falling drop in a pool | 250 |
| 9.1 | Computational domain for the simulation of atomisation | 261 |
| 9.2 | Effect of the coefficient ϵ on the performance of the RCLS | 267 |

| | | |
|------|---|-----|
| 9.3 | Effect of the periodicity of the re-initialisation on the performance of the RCLS | 268 |
| 9.4 | Planes of interface contour extraction | 271 |
| 9.5 | Interface contour in the longitudinal plane | 272 |
| 9.6 | Interface contour in the transverse plane | 273 |
| 9.7 | Simulation of flat sheet breakup with <i>lesRCLSFoam</i> — Build-up of instabilities | 279 |
| 9.8 | Simulation of flat sheet breakup with <i>lesRCLSFoam</i> — Flapping, membrane puncturing and ligament formation | 280 |
| 9.9 | Simulation of flat sheet breakup with <i>lesRCLSFoam</i> — Streamwise ligament pinch-off | 281 |
| 9.10 | Simulation of flat sheet breakup with <i>lesRCLSFoam</i> — Sheet tearing in transverse direction | 282 |
| 9.11 | Simulation of flat sheet breakup with <i>lesRCLSFoam</i> — Sheet tearing in longitudinal direction | 283 |

List of Tables

| | | |
|------|---|-----|
| 4.1 | Simulation settings for the advection of a 2D cross | 68 |
| 4.2 | Simulation settings for the advection of a slotted disk | 70 |
| 4.3 | Simulation settings for the disk in a deformation field | 71 |
| 4.4 | Performance of the AMR on the translation of a 2D cross | 72 |
| 4.5 | Performance of the AMR on the disk in a deformation field | 72 |
| 4.6 | Simulation settings for the Rayleigh-Taylor instability | 77 |
| 4.7 | Simulation settings for the falling drop in a pool | 79 |
| 4.8 | Simulation settings for the phase inversion | 81 |
| 4.9 | Physical properties for the spray calculations with SGS models | 90 |
| 4.10 | Physical properties for the spray calculations without SGS models . . | 90 |
| 4.11 | Geometrical parameters for all the spray calculations | 91 |
| 4.12 | Non-dimensional numbers for all the spray calculations | 91 |
| 4.13 | Smallest scales of the flow field for all simulations without SGS model | 102 |
| 6.1 | Numerical convergence study for the Cartesian meshes | 177 |
| 6.2 | Numerical convergence study for the tetrahedral meshes | 178 |
| 6.3 | Numerical convergence study for the hybrid meshes | 179 |
| 7.1 | Coefficients of the Runge-Kutta scheme SSP(3, 3) of Shu and Osher . | 191 |
| 8.1 | Settings of the droplet transfer algorithm for the static test case . . . | 233 |
| 8.2 | Settings of the droplet transfer for the Rayleigh-Taylor instability case | 235 |

| | | |
|-----|--|-----|
| 9.1 | Non-dimensional numbers associated to the flow simulated | 263 |
| 9.2 | Smallest length scale of the flow field | 263 |
| 9.3 | RCLS parameters for the simulation of atomisation | 269 |

Nomenclature

Roman letters

| | |
|--------------------------------------|---|
| $\hat{\mathbf{a}}$ | Least-squares approximation of the degrees of freedom (polynomial reconstruction) |
| $\hat{\mathbf{n}}$ | Unit vector normal to the interface |
| $\hat{\mathbf{t}}^{(k)}$ | Two independent unit vectors tangent to the interface |
| $\hat{\mathbf{A}}$ | Riemann solver |
| \mathbf{a} | Vector of degrees of freedom (polynomial reconstruction) |
| \mathbf{A}_{n_l} | Numerical flux in the direction \mathbf{n}_l |
| \mathbf{b} | Data vector (polynomial reconstruction) |
| \mathbf{f} | Body forces |
| $\mathbf{F}, \mathbf{G}, \mathbf{H}$ | Vectors of the fluxes in the x, y, z direction |
| \mathbf{f}_Γ | Singular surface force |
| \mathbf{f}_Ω | Volume force approaching \mathbf{f}_Γ |
| \mathbf{f}_{cap} | Source term for the singular capillary forces |
| \mathbf{g} | Gravity force |

| | |
|----------------------------|--|
| \mathbf{n} | Outward unit vector normal to ∂E_i |
| \mathbf{n}_l | Outward unit vector normal to F_l |
| \mathbf{p} | Momentum |
| \mathbf{R} | Vector of source vectors (system of algebraic equations) |
| \mathbf{U} | Vector of conserved variables |
| \mathbf{U} | Vector of solution vectors (system of algebraic equations) |
| \mathbf{u} | Velocity vector |
| \mathbf{u}_t | Tangential velocity vector |
| \mathbf{v} | Vector of coefficient for the Burgers' equation |
| \mathbf{v}_k | Pre-computed vector for the high-order calculation of the gradient |
| \mathbf{X} | Space of the octant $(+, +, +)$ |
| \mathbf{x} | Physical space |
| \mathbf{x} | Spatial coordinate vector |
| \mathbf{x}_Γ | Any point of the interface |
| \mathcal{A} | Matrix associated to the system of algebraic equations |
| \mathcal{A} | Reconstruction matrix |
| \mathcal{A} | Tensor of fluxes for \mathbf{U} |
| \mathcal{A}_τ^\dagger | Effective pseudo-inverse (SVD) |
| \mathcal{B} | Blob in the level set field |
| \mathcal{B} | Oscillation indicator matrix |

| | |
|---|---|
| \mathcal{C} | Cross stress tensor |
| \mathcal{D} | Droplet in the level set field |
| \mathcal{D} | Rate-of-strain tensor |
| \mathcal{I} | Unit tensor |
| \mathcal{J} | Jacobian of the mapping from $\boldsymbol{\xi}$ to \mathbf{x} |
| \mathcal{J}_{N_l} | Jacobian matrix of the vector flux in the direction \mathbf{n}_l |
| \mathcal{J}_Q | Jacobian of the mapping from the octant to the sector S_l |
| $\mathcal{J}_X, \mathcal{J}_Y, \mathcal{J}_Z$ | Jacobian matrices associated to the flux vectors $\mathbf{F}, \mathbf{G}, \mathbf{H}$ |
| \mathcal{L} | Leonard stress tensor |
| $\mathcal{L}_{N_l}, \mathcal{R}_{N_l}$ | Left and right eigenvector matrices of \mathcal{J}_{N_l} |
| \mathcal{O}_{L^p} | Order in the p-norm |
| \mathcal{P} | Stress tensor |
| \mathcal{R} | Reynolds-stress tensor |
| \mathcal{R}_0 | Frame of reference for the mapping to the reference space |
| \mathcal{S} | Stencil for a given cell |
| \mathcal{T} | Residual-stress tensor |
| \mathcal{T}_σ | Tensor of the capillary effects |
| \mathcal{U}, \mathcal{V} | Matrices of orthonormalised eigenvectors (SVD) |
| \mathcal{C} | Cone delimiting the sector S associated to a given face of a cell |
| \mathcal{P}_l | Longitudinal plane cutting the computational domain |

| | |
|------------------------|--|
| \mathcal{P}_t | Transverse plane cutting the computational domain |
| \mathcal{U} | Set of stencils for a given cell |
| $\tilde{\mathbf{a}}$ | Vector of modified degrees of freedom for \mathbf{U} (polynomial reconstruction) |
| $\tilde{\mathbf{a}}$ | Vector of modified degrees of freedom (polynomial reconstruction) |
| \tilde{a} | Modified degrees of freedom (polynomial reconstruction) |
| \tilde{C} | Mollified colour function |
| \hat{F} | Fluxes for a scalar equation in the directions \mathbf{n}_l |
| a | Degree of freedom (polynomial reconstruction) |
| a | Jump of the discontinuous variable f |
| a_Γ | Jump of the discontinuous variable f across the interface |
| A_{n_l} | Numerical flux in the direction \mathbf{n}_l , for a scalar equation |
| C | Composition variable |
| C | Volume fraction, a.k.a. colour function |
| C_Δ | Coefficient of the distribution of LES filter width |
| C_ε, C_μ | Parameters of the sub-grid scale eddy-viscosity model |
| C_{traj} | Centre of the trajectory |
| Ca | Capillary number |
| D | Diameter of the computational domain |
| D | Diameter of the injection channel |
| D | Global computational domain |

| | |
|-------------|---|
| d | Intermediary distance function of the re-distancing algorithm |
| d | Linear weight of a given stencil |
| d | Nozzle outlet diameter |
| d | Spatial dimension of the computational domain |
| d | dilatation |
| D_{eq} | Droplet equivalent diameter |
| D_i | 0-halo sub-domain |
| D_{min} | Diameter of the smallest droplet |
| D_{traj} | Diameter of the trajectory |
| $d_{We=10}$ | Diameter of the droplet at $We = 10$ |
| Dh_i | n-halo sub-domain |
| E | Kinetic energy field |
| E_0 | Targeted cell |
| E_f | Kinetic energy of the filtered velocity fields |
| E_i | i^{th} cell of the computational domain |
| E_j | j^{th} element of the stencil \mathcal{S} |
| f | Free energy density |
| f | Variable discontinuous across the interface |
| F, G, H | Fluxes in the directions x, y, z , for a scalar equation |
| F_l | l^{th} face of the cell E_i |

| | |
|-----------------|---|
| FI_l | l^{th} face of E_i that is an internal face of the mesh |
| g | Acceleration due to gravity |
| G_Δ | LES filter function |
| H | Heaviside function |
| H | Height of the computational domain |
| h | Artificial thickness of the interface |
| h | Grid spacing |
| h | Height of the nozzle outlet |
| H_ε | Heaviside function mollified over a distance ε |
| i, j | Spatial index |
| ID | Integrand for the calculation of the terms of \mathcal{B} |
| IS | Oscillation indicator of a given stencil |
| j_{max} | Number of element in the stencil \mathcal{S} |
| K | Number of degrees of freedom (polynomial reconstruction) |
| k_β | Weight of the Gaussian quadrature for the β^{th} Gaussian point |
| k_r | Residual kinetic energy |
| L | Characteristic length |
| L | Length of the injection channel |
| l, k | Level of grid refinement |
| $L(\varphi)$ | Time evolution of φ |

| | |
|-------------------|---|
| L^p | p-norm |
| L_C | Characteristic length of the liquid core |
| L_b | Breakup length |
| L_i | Number of faces of E_i |
| La | Laplace number |
| M | Momentum flux ratio |
| m | Mass |
| Ma | Mach number |
| N | Number of cells in one dimension |
| $N_{\mathcal{B}}$ | Number of cells in the blob \mathcal{B} |
| N_C | Minimum number of cells in the liquid core |
| $N_{\mathcal{D}}$ | Number of cells in the droplet \mathcal{D} |
| N_{β} | Number of Gaussian points |
| N_c, N_{cells} | Number of cells of the mesh |
| N_{FI} | Number of faces of a given cell that are internal faces of the mesh |
| N_{S_i} | Number of sectoral stencils of the cell E_i |
| N_s | Number of steps between two re-initialisation (TCLS) |
| N_T | Number of triangles composing FI_l |
| Oh | Ohnesorge number |
| p | Exponent of the oscillation indicator |

| | |
|------------|---|
| p | Polynomial interpolation of the solution |
| p | Pressure |
| p_d | Modified pressure |
| p_{WENO} | WENO polynomial interpolation of the solution |
| Q | Q-criterion: second invariant of the velocity gradient tensor |
| Q | Volumetric flow rate |
| q | Mass transfer from one phase to the other |
| r | Order of the polynomial reconstruction |
| R_1, R_2 | Principal radii of curvature |
| Re | Reynolds number |
| S | Shock speed (Riemann problem) |
| S_{l_i} | Sub-sector of S_l |
| S_l | Sector associated to the face FI_l |
| T | Thickness of the computational domain |
| t | Thickness of the nozzle outlet |
| t | Time |
| t_{calc} | Duration of the calculation |
| t_R | Time at which the droplet transfer occurs |
| U | Characteristic velocity |
| u' | Normal velocity in the case of phase change |

| | |
|-------|---|
| u_p | Magnitude of the spurious currents |
| u_x | Velocity component along the axis \vec{x} |
| u_y | Velocity component along the axis \vec{y} |
| V | Interface velocity |
| V | Volume |
| w | WENO weight of a given stencil |
| We | Weber number |
| x | Spatial coordinate |

Greek letters

| | |
|-----------------------------|---|
| $\alpha_{i,k}, \beta_{i,k}$ | Coefficients of a general Runge-Kutta scheme |
| ξ | Reference space |
| $\Delta\Gamma$ | Area representing a portion of the interface |
| $\Delta\Omega$ | Small volume of thickness h bounding $\Delta\Gamma$ |
| ΔC | Difference in volume fraction |
| Δm | Mass error |
| Δp | Pressure difference across the interface |
| Δx | Grid size |
| Δ | LES filter width |
| δ | Dirac delta function |
| δ | Thickness of the interface (CLS) |

| | |
|----------------------|--|
| δ_Γ | Distribution concentrated on the interface |
| δ_ε | Dirac delta function mollified over a distance ε |
| ϵ | Coefficient controlling the thickness of the interface (CLS) |
| η | Kolmogorov micro length scale |
| Γ | Interface |
| γ | Material property |
| κ | Interface curvature |
| λ | Characteristic speed (Riemann problem) |
| λ | Second coefficient of viscosity |
| λ_l | Limiter factor |
| Λ_{N_i} | Diagonal matrix of eigenvalues of \mathcal{J}_{N_i} |
| μ | Dynamic viscosity |
| μ_r | Eddy-viscosity of the residual motions |
| ν | Kinematic viscosity |
| Ω | Computational domain |
| Ω | Region surrounding the immersed boundary |
| Ω^-, Ω^+ | Two domains separated by the interface |
| ∂E_i | Boundary of the cell E_i |
| ϕ | Level set function |
| ϕ_Γ | Value of the level set function at the interface |

| | |
|---------------------|---|
| ϕ_k | k^{th} basis function (polynomial reconstruction) |
| Ψ | Chemical potential of the composition variable |
| ψ | Conservative level set function |
| ψ | Deformation field function |
| ψ_Γ | Value of the conservative level set function at the interface |
| ψ_k | k^{th} monomial (polynomial reconstruction) |
| ρ | Density |
| Σ | Diagonal matrix of singular values of \mathcal{A} |
| σ | Surface tension |
| σ_i | i^{th} Singular value of \mathcal{A} |
| τ | Artificial time step (re-distancing/re-initialisation procedure) |
| τ | Tolerance (SVD) |
| τ_c | Characteristic time scale of convection |
| τ_s | Characteristic time scale of surface tension effects |
| τ_v | Characteristic time scale of viscous effects |
| $\tilde{\varphi}$ | Virtual field (TCLS) |
| ε | Small positive number involved in the calculation of the WENO weights |
| ε | Very small value of volume fraction ($\varepsilon \ll 1$) |
| ε_{L^p} | Error measured with the p-norm |
| φ | Truly conservative level set function |

| | |
|----------------------------|--|
| $\varphi_{\mathcal{D}min}$ | Threshold of volume fraction (droplet selection) |
| φ_{Lmin} | Threshold of volume fraction (blob selection) |
| ζ | Scale of the smallest liquid structure |

Subscript indices

| | |
|---------------|--|
| 1 | Relating to the heavy phase |
| 2 | Relating to the light phase |
| β | Relating to the β^{th} Gaussian point |
| ξ | Relating to the reference space |
| Γ | Relating to the interface |
| \hat{n} | Relating to the direction $\hat{\mathbf{n}}$ |
| \mathbf{x} | Relating to the physical space |
| \mathcal{B} | Relating to the blob |
| \mathcal{D} | Relating to the droplet |
| CFD | Relating to the numerical results |
| D | Deviatoric part of a tensor |
| E_i, E_j | Relating to the cells E_i, E_j |
| EXP | Relating to the experimental results |
| <i>final</i> | Relating to the final time step |
| <i>gas</i> | Relating to the gas phase |
| i, j | Spatial index |

| | |
|----------|---|
| j | Relating to the j^{th} cell in the stencil |
| k | Relating to the k^{th} degree of freedom |
| l | Relating to the l^{th} face of a given cell |
| $l = 6$ | Relating to the level of refinement: 6 |
| $l = 7$ | Relating to the level of refinement: 7 |
| liq | Relating to the liquid phase |
| m | Relating to the m^{th} stencil |
| n_l | Relating to the direction \mathbf{n}_l |
| $nozzle$ | Relating to the nozzle location |
| oil | Relating to the oil phase |

Superscript indices

| | |
|-------|---|
| $+$ | Relating to the Ω^+ domain |
| $+$ | Relating to the limit of a quantity at the face F_l inside the cell E_i |
| $-$ | Relating to the Ω^- domain |
| $-$ | Relating to the limit of a quantity at the face F_l outside of the cell E_i |
| (m) | Relating to the m^{th} stencil |
| $g+$ | Relating to ghost cells in the Ω^- domain (ghost extension of Ω^+) |
| $g-$ | Relating to ghost cells in the Ω^+ domain (ghost extension of Ω^-) |
| n | Relating to the n^{th} iteration |

Other symbols

| | |
|-------------------------------|--|
| $(\cdot)'$ | Relating a geometrical element of the mesh to its counterpart in the reference space |
| $(\cdot)'$ | Residual component of a quantity |
| $[\cdot]_{\Gamma}$ | Jump condition operator across the interface |
| $\llbracket \cdot \rrbracket$ | Implicit discretisation of a term |
| $\overline{(\cdot)}$ | Average of a quantity |
| $\overline{(\cdot)}$ | Filtered quantity (LES) |

Acronyms

| | |
|--------|--|
| ACLS | Accurate Conservative Level Set method |
| ADER | Arbitrary high order schemes using DERivatives |
| ALE | Arbitrary-Lagrangian-Eulerian method |
| AMR | Adaptive Mesh Refinement |
| BI | Boundary Integral method |
| CFD | Computational Fluid Dynamics |
| CFL | Courant-Friedrichs-Lewy condition |
| CLS | Conservative Level Set method |
| CLSVOF | Coupled Level Set - Volume Of Fluid method |
| CN | Crank-Nicholson scheme |
| CNRS | Centre National de la Recherche Scientifique |
| CPU | Central Processing Unit |

| | |
|-----------|--|
| CSF | Continuum Surface Force method |
| CSS | Continuum Surface Stress method |
| DDR | Defined Donating Region |
| DNS | Direct Numerical Simulation |
| ELVIRA | Efficient LVIRA |
| ENO | Essentially Non-Oscillatory scheme |
| FCT | Flux-Corrected Transport scheme |
| FLAIR | Flux Line-segment model for Advection and Interface Reconstruction |
| GAMG | Generalised geometric-Algebraic Multi-Grid solver |
| GFM | Ghost Fluid Method |
| GHG | GreenHouse Gas |
| Hi-Fi LES | High-Fidelity Large Eddy Simulation |
| IC | Initial Condition |
| IIM | Immersed Interface Method |
| ILES | Implicit Large Eddy Simulation |
| KH | Kelvin-Helmoltz instability |
| LBM | Lattice Boltzmann Method |
| LES | Large Eddy Simulation |
| LGA | Lattice-Gas Automata |
| LS | Level Set method |

| | |
|-------|---|
| LVIRA | Least square Volume-of-fluid Interface Reconstruction Algorithm |
| MAC | Marker-And-Cell method |
| MCLS | Mass-Conserving Level Set method |
| ME | Momentum Equation |
| MMD | Mass Median Diameter |
| NS | Navier-Stokes equations |
| OEEVM | constant coefficient One-Equation Eddy-Viscosity Model |
| PCG | Pre-conditioned Conjugate Gradient solver |
| PDE | Partial Differential Equation |
| PDF | Probability Density Function |
| PFM | Phase Field Method |
| PLIC | Piecewise Linear Interface Calculation |
| PLS | Particle Level Set method |
| PROST | Parabolic Reconstruction Of Surface Tension |
| RANS | Reynolds-Averaged Navier-Stokes equations |
| RK | Runge-Kutta scheme |
| RLSG | Refined Level Set Grid method |
| RT | Rayleigh-Taylor instability |
| SGS | Sub-Grid Scale |
| SLIC | Simple Line Interface Calculation |

| | |
|------|--|
| SMD | Sauter Mean Diameter |
| SPH | Smoothed Particle Hydrodynamics |
| SSP | Strong-Stability Preserving scheme |
| SVD | Singular Value Decomposition |
| TCLS | Truly Conservative Level Set method |
| TK | exact Transport equation for the residual Kinetic energy |
| TVB | Total Variation Bounded scheme |
| TVD | Total Variation Diminishing scheme |
| VOF | Volume Of Fluid method |
| WENO | Weighted Essentially Non-Oscillatory scheme |

Chapter 1

Introduction

Atomisation is the process that transforms bulk liquid into sprays [122]. This process plays an important role in a broad range of industries and sciences such as: aeronautics (rockets and aircraft), automotive engineering, pharmaceutical, power generation, petro-chemical, manufacturing, agriculture and meteorology.

Although atomisation is widely used and drives the performance of many systems, the characteristics of the spray produced (e.g., droplet size and droplet velocity distributions) are still poorly predicted. This is particularly true for aero-engines which generally rely on air-blast atomisers to inject the kerosene in combustion chambers.

As the prediction of fuel sprays in gas turbines is of critical importance to maximise the combustion efficiency and reduce the aviation emissions, aero-engine manufacturers invest in the generation of numerical methods to model the injection process.

As part of such a research programme, this work aims at providing the aeronautical industry with a modelling tool to simulate the fuel injection.

1.1 The aeronautical application

The kerosene is generally injected in combustion chambers as an annular liquid sheet sheared on either side by a faster co-flowing gas stream. This sheet undergoes a series of instabilities (longitudinal and transverse) which lead to the fragmentation of the liquid bulk into liquid structures that further disintegrate into droplets (see Figure 1.1). This initial process of the atomisation is called the primary breakup and occurs in the vicinity of the injection point (see Figure 1.1).

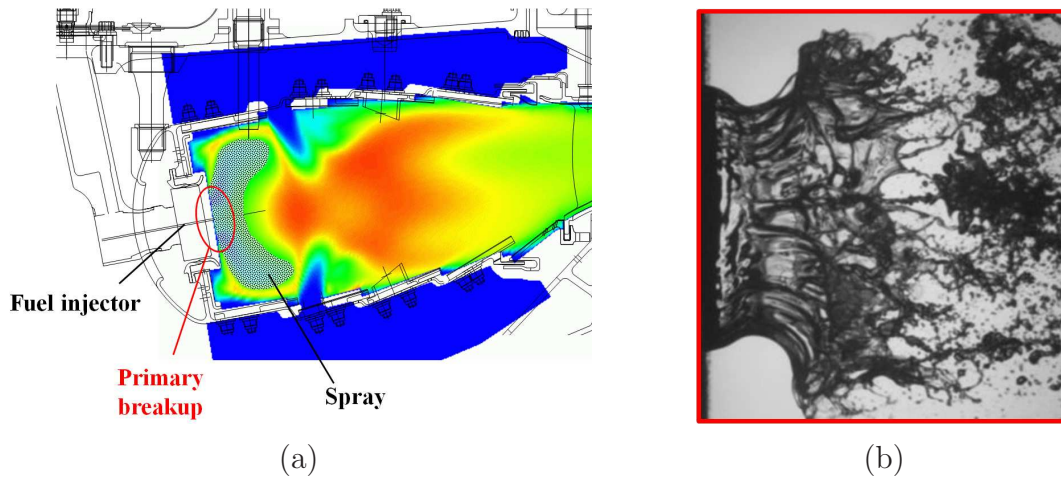


Figure 1.1: Atomisation in a combustion chamber: (a) Cartoon of the kerosene spray (slice) overlaying the temperature contours in a combustion chamber (courtesy of Rolls-Royce plc); (b) Close-up in the region of the primary breakup [37]

The formation of the fuel spray also involves the transport of the droplets produced by the primary breakup, their disintegration into smaller drops and the coalescence of liquid structures. This is the secondary breakup which operates further away from the bulk liquid.

The mechanisms of the primary breakup initiate the atomisation process, control the extent of the liquid core and provide initial conditions for the secondary breakup in the disperse flow region.

1.2 Motivations

Due to the increasing concern about global warming and more generally the human impact on the environment, governments have recently produced more stringent emission standards for the aeronautical industry. As the production of NO_X and CO_2 in gas turbines is affected by the fuel-air mixing in combustion chambers, aero-engine manufacturers expect to reduce the emissions of greenhouse gas (GHG) through the optimisation of the fuel injection. As aircraft engines generally operates under a wide range of conditions, optimising the fuel-gas mixing over the entire flight envelope is extremely difficult.

The fuel-gas mixing is primarily driven by the atomisation which involves both the initial fragmentation of the bulk liquid into droplets (primary breakup) and the transport and further fragmentation of the drops (secondary breakup). Whereas the secondary breakup is fairly well predicted by the current numerical methods, the accurate simulation of the primary breakup remains one of the toughest challenge in computational fluid dynamics (CFD).

However, the potential of the numerical approach to study and simulate the liquid fragmentation is high. With an accurate modelling capability, the aeronautical industry would not have to rely solely on comprehensive experimental test campaigns and the design of efficient devices would be cheaper. Also, with the recent progress in experimental measurements of the multiphase flows, the combination of the numerical tool with the experimental approach would allow the manufacturers to improve the combustion efficiency, reduce the emissions of GHG and lower the fuel consumption.

1.3 Approaches to study the primary breakup

The understanding of the mechanisms controlling the primary breakup is currently limited because of the difficulty in observing and measuring flow properties in the

dense spray region. The most obvious means of studying the primary breakup, and historically the first one, is the experimental study. The recent progress in high-speed camera technology has allowed major advances in the description of the breakup process and the structure of the spray.

Scientists have also used linear stability analysis to describe the phenomenon, in particular Rayleigh tackled this approach in 1879 [192, 193]. This theoretical framework has given useful insight on simple configurations. Unfortunately, many features of the breakup process are dominated by non-linear phenomena and cannot be tackled by even weakly non-linear stability analysis. However, the final stages of liquid fragmentation — the secondary breakup — even though highly non-linear have been described statistically using scale invariance.

Finally, the numerical characterisation of the primary breakup has grown in popularity in the past decades. It has proven to be a very difficult problem due to the wide variety of time and length scales associated with the atomisation. Here, the challenge consists of capturing the detailed physics of individual breakup events while representing the complex geometries of the injection devices.

1.4 Numerical modelling of atomisation

1.4.1 Numerical framework

The breadth of turbulence scales calculated drives the choice of the numerical framework: from the calculation of all the scales of motions with direct numerical simulations (DNS) to the direct representation of the energy containing motions (and therefore the modelling of the effect of the smaller scale motions) in large eddy simulations (LES) to finally the complete modelling of the turbulence with the Reynolds-averaged Navier-Stokes (RANS) approach.

The numerical study of atomisation has been tackled at various levels of idealisation: RANS, LES with and without (Euler/Lagrange and Euler/Euler formulations)

interface description and DNS. However, only the LES and DNS frameworks with interface description can provide valuable insight into the fundamental physics of the primary breakup. This work has then focussed on this type of approach.

1.4.2 Interface description

The description of the interface is necessary to study numerically the underlying physics of the primary breakup. Various interface description methods have been developed for the simulation of multiphase flows and the most popular ones can be categorised into two groups: the methods describing the interface explicitly (moving mesh and front tracking) and implicitly (volume of fluid and level set). In particular, implicit interface description methods handle the changes of topology automatically and offer great potential for the simulation of the atomisation.

The main challenge in developing an interface description method is to produce an implicit technique that conserves mass (like volume of fluid) while predicting accurately the interface location (like level set).

1.4.3 Treatment of singularities

The simulation of multiphase flows with interface description generally involves an immersed boundary — the phase interface — moving in a fixed grid. This boundary is the locus of surface tension forces and material discontinuities that can be expressed as jump conditions (see Section 3.1.3). Even though, the Weber number We , is typically very high at aero-engine injection conditions, the breakup occurs at small scales where We is small and the capillary forces dominant [146].

Although numerically challenging, the accurate modelling of surface tension and material discontinuities is crucial to the successful simulation of primary breakup. There are essentially two methods to handle numerically these singularities: the widely used continuum surface force (CSF) method popularised by Brackbill et al. [26] and the more recent ghost fluid method (GFM) of Fedkiw et al. [55].

1.5 Aim of the present work

Building upon the recent improvements of the numerical approach to model the primary breakup, this work has focussed on the generation of an efficient modelling capability to simulate the atomisation process in the combustion chambers of industrial gas turbines.

Real engine problems are characterised by the complexity of the mechanical boundary conditions, the very large breadth of length and time scales involved in the atomisation process and the limited amount of resource available for the whole numerical study.

As a result, to be relevant to the industry, a modelling capability has to be developed for unstructured grids. Indeed, with such an approach, the geometric details of the injection device would be faithfully represented in a timely manner, as unstructured meshes are generally generated faster. In addition, due to the limited amount of resources available, it is essential to base the modelling tool on numerical methods that provides the best trade-off in terms of accuracy (mass conservation and interface location) versus computational cost. Finally, the breadth of length scales governing the spray can only be handled through the use of a sub-model for the prediction of the primary breakup. Therefore, it is necessary that the modelling capability outputs the droplet boundary conditions required by the combustion codes to transport the spray in the combustion chamber.

In order to satisfy the above requirements, the work presented in this document has focussed on:

- The generation of high-order accurate numerical schemes for unstructured meshes
- The development of an efficient numerical method that transports the interface accurately while conserving the mass.
- The implementation of a modelling tool that outputs the characteristics of the

droplets produced by the atomisation process.

1.6 Outline of the thesis

This thesis is articulated in two main parts. In the first part, from Chapter 2 to Chapter 4, the document provides some background on the physics of the primary breakup, reviews the existing numerical methods to simulate multiphase flows and compares the available multiphase codes.

Then, in Chapter 5, building upon atomisation modelling tools demonstrated on idealised configurations, we detail the methodology adopted to produce a capability to simulate the fuel injection in gas turbines. This chapter operates as a transition between the two parts of the thesis.

In the second part of this document, we describe the building blocks of the novel modelling capability produced. In particular, Chapter 6 presents the high-order numerical scheme developed for unstructured meshes and Chapter 7 depicts the mass-conservative numerical method produced to transport the liquid. Then, in Chapter 8 we describe the additional components of the modelling capability, those making the numerical tool readily applicable to the simulation of liquid sheet breakup. Finally, in Chapter 9, the modelling tool is demonstrated on the simulation of the primary breakup.

The thesis concludes with Chapter 10, where the achievements of this project are summarised and some follow-on research topics are suggested.

Chapter 2

Physics of primary breakup

The fundamentals of liquid fragmentation have been studied since the beginning of the nineteenth century and, although its detailed understanding is limited, scientists have made significant progress in the physical description of the phenomenon.

This chapter first describes the fundamentals of the physics of multiphase flows and then presents the principal scientific findings for the breakup phenomenon. In particular, the current physical description of the primary breakup is given in Section 2.3 and Section 2.4 for round liquid jets and flat sheets respectively.

2.1 Fundamental forces and dimensional analysis

Neglecting body forces, three forces are involved in multiphase flows: capillary forces, inertial forces and viscous forces. The balance of these three forces drives the behaviour and the shape of the interface.

Multiphase flows (in particular atomisation) introduce a broad variety of length and time scales and the relative importance of the physical phenomena involved varies according to the scale considered (e.g. capillary effects dominate at small scales and inertial effects at large scales). It is therefore necessary to introduce non-dimensional parameters in order to identify the prevailing physics. For a given length

scale L , three time scales can be estimated, relating respectively to: convection, surface tension effects and viscous effects. By comparing these time scales, three non-dimensional parameters can be produced:

Reynolds number (Re): Inertial forces relative to viscous forces,

Weber number (We): Fluid inertia relative to its surface tension,

Ohnesorge number (Oh): Viscous forces relative to surface tension.

$$Re = \frac{\tau_v}{\tau_c} = \frac{\rho UL}{\mu} \quad We = \left(\frac{\tau_s}{\tau_c}\right)^2 = \frac{\rho U^2 L}{\sigma} \quad Oh = \frac{\tau_s}{\tau_v} = \frac{\mu}{\sqrt{\rho\sigma L}} \quad (2.1)$$

where τ_c , τ_s and τ_v are the characteristic time scales of respectively the convection, the surface tension effects and the viscous effects.

2.2 Early research interest

Fundamental to the physical understanding of multiphase flows, the surface tension acts as a singular surface force on the liquid-gas interface and tends to minimise the surface energy by reducing the interface area. This phenomenon was discovered by Laplace and Young in 1805 [283].

Savart was the first to show significant interest in liquid fragmentation. In 1833, he conducted experimental studies discussing the formation and the breakup of planar jets [213–216]. In particular, [215] and [216] are concerned with jet impacting a rotating disc and [214] describes the collision of two jets forming a liquid sheet. In the latter experiment Savart observed the growth of undulations at the jet surface leading to the breakup of the jet. Later on, Plateau interpreted the effect of surface tension on these waves [179].

In 1879, Rayleigh formulated the effect of surface tension on breakup [192, 193]. Considering a small sinusoidal perturbations at the surface of a liquid jet, Rayleigh

demonstrated that the fastest growing wavelength dominates the breakup. He then derived the drop size produced using mass conservation. The success of his approach was confirmed by the validation of his calculations against Savart's experiments.

2.3 Round liquid jet breakup

2.3.1 Jet breakup regimes

The onset of the breakup is related to the growth of small disturbances on the liquid surface due to the interaction of the liquid jet with the ambient gas. The modification of the jet velocity affects the relative influence of liquid inertia, surface tension, viscous forces and aerodynamic forces on the jet.

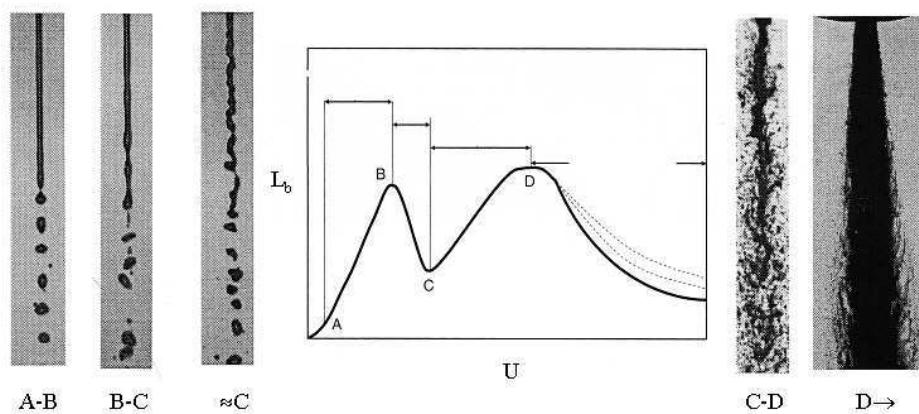


Figure 2.1: Plot of the breakup length vs. jet velocity and the corresponding breakup regimes for a round liquid jet resulting in a quiescent gas [124]

By plotting the breakup length against jet velocity four breakup regimes have been identified (see Figure 2.1):

Rayleigh regime (A–B): In this regime, the capillary forces are responsible for the breakup. The droplets produced have a diameter larger than the jet and

the breakup occurs many jet diameters downstream of the injection plane. This regime is describe by Rayleigh's theory [192].

1st wind-induced (B–C): Here, the aerodynamic forces take over the capillary forces. This leads to the reduction of the breakup length and results in droplets of the order of the jet diameter. This regime is describe by Weber's theory [270].

2nd wind-induced (C–D): For this regime, the axial symmetry is lost and droplets are peeled off from the liquid core. The jet flow starts to be turbulent and the aerodynamic forces produce ligaments on the surface of the liquid core. These ligaments breakup into droplets smaller than the jet diameter.

Atomization (D →): At these conditions, the jet flow is fully turbulent. This turbulence in the liquid phase initiates perturbations of the jet interface further amplified by the aerodynamic forces. The drop formation starts at the nozzle exit and results in droplets much smaller than the jet diameter. In this regime, cavitation can occur in the injection passage and may affect the liquid fragmentation. The effect of cavitation is multiple: it can either increase the turbulence level (acceleration of the atomisation process) or laminarise the jet by limiting the friction on passage walls (remove the turbulent boundary layer).

2.3.2 Spray structure

Faeth et al. observed the structure of sprays on non-evaporating round pressure-atomised sprays in still gas [54]. Such a simple configuration allows easy identification of the fundamental features of sprays. Faeth et al. justify the non-evaporating assumption by noting that the dense spray region of combusting sprays involves cool portions of the flow where the rates of heat and mass transfer are modest [54].

In this configuration, the liquid phase appears in two forms: the liquid core which

extends continuously the liquid phase from the injection hole and the dispersed flow beyond the surface of the liquid core.

- The liquid core is similar to the potential core of a single phase turbulent jet but is in general much longer [33].
- The disperse flow region is composed of blobs of liquids and droplets detached from the liquid core. This region starts with the multiphase mixing layer in the vicinity of the liquid core and evolves into the dilute spray (lower volume fraction: below 0.1%).

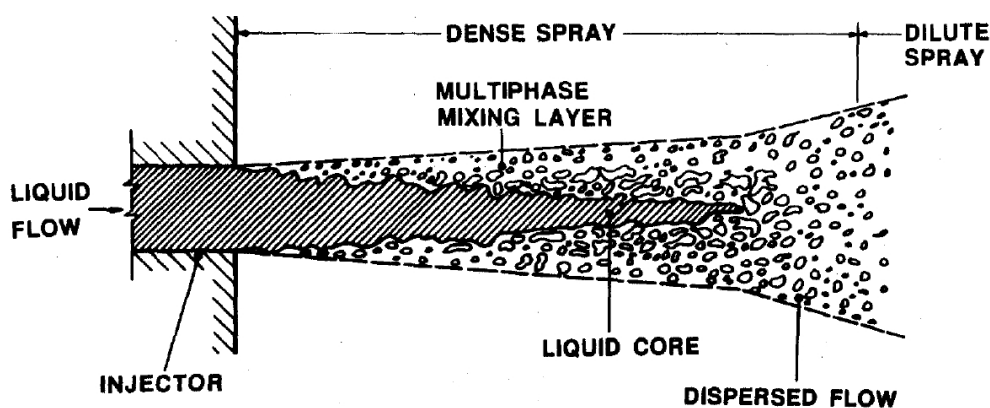


Figure 2.2: Structure of a pressure-atomised spray in the atomisation regime [54]

The spray is commonly divided in two consecutive regions: the dense spray region — associated with the presence of the liquid core — and the dilute spray region corresponding to a lower liquid volume fraction.

The atomisation process involves the primary breakup at the liquid core surface followed by the secondary breakup of liquid structures in the disperse flow. For a round jet the primary breakup occurs essentially via the breakup of large droplets at the end of the liquid core or the pinch off of ligaments formed at the core surface. The resulting liquid structures in general have irregular shapes and may undergo further fragmentation (secondary breakup) depending on the relative importance of the forces acting on them.

2.3.3 Properties of round jet sprays

The properties of such sprays are influenced by a wide range of parameters such as the nozzle exit flow conditions, occurrence of cavitation in the nozzle, velocity profiles, turbulence level in both phases, etc . . . In particular, Faeth et al. observed that the droplet size after the primary breakup and the mixing rate between the two phases are very dependent on the phase density ratio, the flow development and the turbulence level at the nozzle exit [54].

From the analysis of experimental data, Faeth et al. also reported that the droplet sizes, after the primary breakup as well as after the secondary breakup, follow the universal root-normal distribution characterised by a single moment: $MMD/SMD = 1.2$ due to Simmons [233].

2.3.4 Ligament formation

As illustrated in the four different regimes of round jet breakup, the behaviour of the jet depends strongly upon the relative liquid-gas velocity. Whereas the whole liquid column is deformed at low relative velocities, leading to the formation of bags and rims; the liquid jet is merely peeled off at its surface at higher relative velocities resulting in the formation of ligaments. The latter regime being more relevant to aero-engine injection, the ligament formation is detailed below.

Fuster et al. [65] summarise the action of primary breakup at high velocities by two phenomena: the detachment of small ligaments (and droplets) from the jet and further downstream the breakup of large droplets due to the growth of large scale instabilities on the jet surface.

While linear stability analysis and weakly non-linear theories — reviewed in [52] — describe with success the growth of longitudinal (Kelvin-Helmoltz) instabilities along the jet, the formation of ligaments is not well understood. Two mechanisms have been proposed to explain the origin of ligaments:

- Faeth et al. [54] hypothesised that if a sufficient level of turbulence is attained

in the liquid phase, upstream of the nozzle, it could deform the interface and lead to the breakup. The ligament would therefore be formed thanks to a turbulent eddy with sufficient energy to overcome surface tension.

- Marmottant and Villermaux [146] modelled the flow around the jet interface by a two-phase mixing layer. In this framework, they demonstrated (using stability analysis) that a sufficient relative velocity between phases leads to the growth of a series of instabilities on the jet surface and then results in the formation and the breakup of ligaments.

Interface deformation by turbulence in the liquid phase The hypothesis of Faeth et al. [54] resulted from the analysis of the experimental work of [92, 93, 142, 206, 207, 274–277] on round jets breaking up in a quiescent gas. In particular, they noted that:

- The droplet sizes after primary breakup are strongly related to both the level of flow development and the intensity of the turbulence at the nozzle exit,
- The breakup is associated with the presence of a turbulent boundary layer in the liquid phase, at the nozzle outlet,
- The liquid phase properties govern almost entirely the primary breakup properties for large density ratios (> 500),
- The onset of turbulent primary breakup is strongly related to the laminar-turbulent transition in the injector passage.

In addition, basing their analysis on time scale considerations, Faeth et al. argue that the droplets produced near the nozzle exit (i.e. at the onset of the primary breakup) are the smallest that can be formed [54]. The size of the smallest drop is then given by the size of the “critical turbulent eddy” which has just enough kinetic energy to provide the surface energy to form a drop. When the “critical turbulent

eddy” reaches the limit of the inertial range of the turbulence, the smallest droplets are then comparable to the Kolmogorov micro length scale.

Ligament formation through a series of instabilities The assisted atomization of a liquid jet comes with the formation of ligaments at the jet surface (regardless of the geometry) conditioned upon sufficient shear from the gas stream. These ligaments are regularly spaced in the transverse direction and their number increases with the gas velocity.

Marmottant and Villermaux describe the ligament formation and breakup in a sequence of steps for a slow liquid jet sheared by a co-axial gas stream [146]. At first, the two initially parallel streams of different velocities are subject to the Kelvin-Helmoltz instability which leads to the formation of axisymmetric undulations on the liquid interface. The shape of the wave crests become singular as this primary instability grows and develop into liquid sheets bounded by a rim.

Because the bulk liquid velocity is lower than the travelling speed of the longitudinal undulations at the jet surface, the liquid interface is accelerated perpendicular to itself, alternatively towards the liquid and the gas phase. When the acceleration is oriented towards the denser phase, the interface undergoes a Rayleigh-Taylor instability [128, 194, 249] which results in the growth of azimuthal undulations (first hypothesised by Villermaux and Clanet in [266]). Bremond’s findings corroborate this hypothesis in [27].

Then, the development of the longitudinal and transverse instabilities leads to the formation of corrugations at the nodes of their undulations. Initially, these crests are accelerated with respect to the bulk flow with little deformation until the gas drag is sufficient to pull them away from the jet and increase their aspect ratio. Finally, after stretching in the gas stream, these ligaments detach from the liquid bulk by pinching off their base. With a nearly cylindrical shape, the ligaments are per se sensitive to the Plateau-Rayleigh instability and thus further fragment into blobs of liquid.

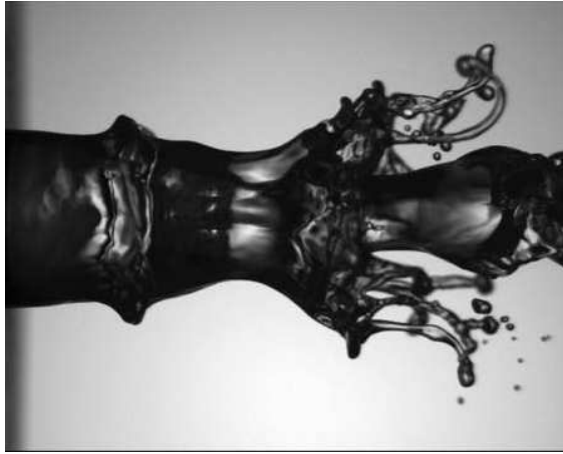


Figure 2.3: Photo of the assisted disintegration of a round liquid jet in a co-axial gas stream [146]

2.4 Liquid sheet breakup

In the planar configuration, the liquid is injected between the two parallel walls of the injector such that the height of the slit is much smaller than its width in order to avoid boundary effects.

Although aero-engine injectors usually show an axisymmetric configuration, this section is of particular relevance to the aeronautical industry as Meyer and Weihs have demonstrated that the mechanisms are identical for planar and axisymmetric geometries in the limit of a thin annulus [153]. Meyer and Weihs [153] have also demonstrated that an annular liquid jet behaves like a full round liquid jet in the limit of a thick annulus.

A very detailed literature review for the liquid sheet breakup can be found in [37]. In this section we will simply summarise the main scientific findings in that field.

2.4.1 Natural disintegration

The first linear stability calculations for an inviscid 2D liquid sheet in quiescent air were conducted in the 1950's: [75, 235, 250, 279]. Hagerty and Shea [75] observed experimentally the sinuous and varicose modes of the liquid sheet interface predicted by the theoretical analysis (see Figure 2.4).

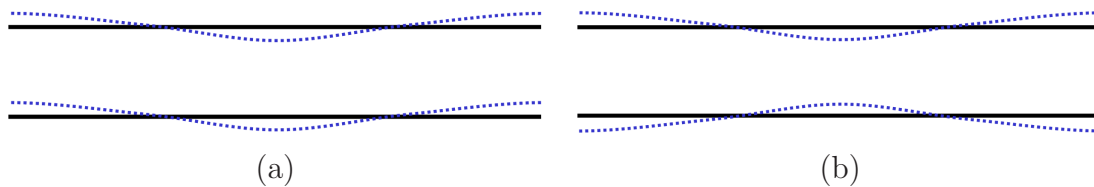


Figure 2.4: The two modes growing along an inviscid liquid sheet: (a) Sinuous mode; (b) Varicose mode

Crapper et al. [39], Li and Tankin [130] and Ibrahim [98] studied the influence of the viscosity on the behaviour of the liquid sheet. In particular, their work suggests that the effect of viscosity on the instability is complex and that it can broaden the range of wave number corresponding to an unstable liquid sheet.

The large deformations of the sheet occur in the sinuous mode predicted by the linear stability theory. However, in order to explain the sheet breakup Jazayeri and Li [105] and Rangel and Sirignano [191] had to conduct non-linear stability analysis. Their studies relate the breakup to the pinching of the sheet in its thickness through the effects of the harmonics of the oscillation modes.

2.4.2 Assisted non-linear disintegration

The first experimental study of an assisted disintegration was carried on round liquid jet configuration by Nukiyama and Tanasawa [161]. This study of a jet sheared by a co-flowing gas stream highlighted the strong correlation between the drop size and the gas stream velocity. The equivalent work on a planar jet was done by Rizk and

Lefebvre [203] and Arai and Hashimoto [8] who also noted, for this configuration, the important impact of the gas velocity on the SMD and breakup length.

Because the initial deformations of the jet depend strongly on the gas phase, the presence of a surrounding gas flow changes fundamentally the mechanisms of the breakup. Cousin and Dumouchel [38] and Barreras [12] produced linear and non-linear stability analysis specifically for this case and their conclusion was twofold. First, the same types of mode — sinuous and varicose — grow on the surface of a liquid sheet sheared by a co-flowing gas; and secondly, the gas and liquid viscosities and the non-linearity of the interface deformation cannot be neglected in the analysis.

A major advance on the understanding of the sheet breakup mechanisms was made in the 1990's by Stapper et al. [236, 237] and Mansour and Chigier [140, 141]. Noting that the breakup is influenced by a wide variety of parameters, these authors investigated the breakup regimes obtained for a broad range of conditions. Specifically, their studies consider the influence of the following physical variables: surface tension, viscosity, thickness of the sheet ($200\mu m$ to $1mm$), velocities in the liquid (1 to $5ms^{-1}$) and the gas (15 to $60ms^{-1}$).

From this set of experiments, Stapper and Samuelsen [236] observed experimentally two regimes (see Figure 2.5):

Cellular breakup: characterised by longitudinal and transverse undulations of similar wavelengths. The combination of these two modes of perpendicular undulations lead to the formation of “liquid cells” which eventually break up.

Stretched streamwise ligament breakup: characterised by a much faster growth of the transverse instability than the longitudinal one. For this regime, which appears for low liquid velocities, the breakup starts closer to the injector exit.

The two regimes also differ by the initial direction of fragmentation: it starts along the longitudinal direction for the stretched streamwise ligament breakup, whereas it starts along the transverse direction for the other regime. However, in both

instances, the transverse instability and the co-flowing shearing gas stream leads to the formation of ligaments at the top of the crests. Mansour and Chigier [140] identified a third breakup regime apparent at low liquid velocities by plotting the fundamental mode of the sheet undulations against the liquid exit velocity. However, this regime was less studied by the authors.

Experimental studies [136, 140, 237] suggested that the stretched streamwise ligament breakup provides the best atomisation (large deformation of the sheet, wide spray angle and smaller droplets). The mechanism of this regime can be summarised in a series of steps. At first the liquid sheet undergo a Kelvin-Helmoltz instability, leading to the quick growth of the sinuous mode, which results in the flapping of the sheet. This growth carries all along the jet atomisation leading to the scattered shedding of droplets. Downstream of the zone of flapping, transverse undulations appear (see [266]) leading to the formation of streamwise ligaments under aerodynamic forces. These ligaments then stretch and eventually fragment into relatively large droplets [237]. The liquid blobs produced eventually undergo a secondary breakup.

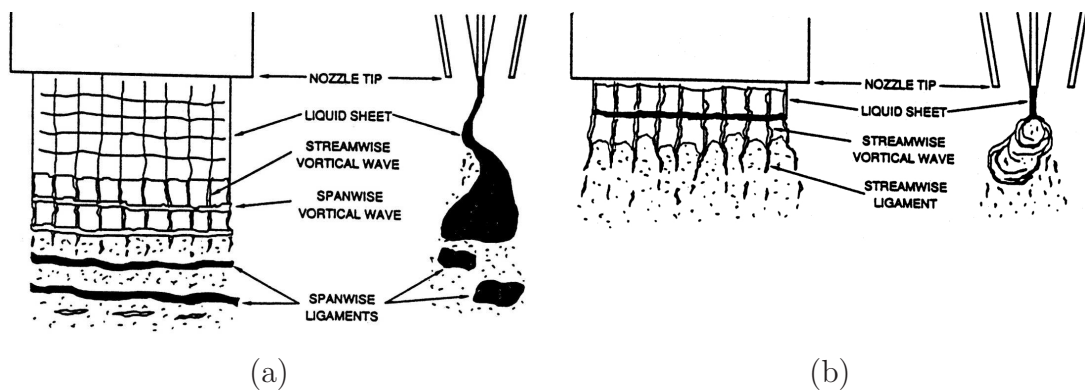


Figure 2.5: The two sheet breakup regimes observed by Stapper and Samuelsen [236]: (a) Cellular breakup; (b) Stretched streamwise ligament breakup

In the 2000's various authors have tackled the parameterisation of the primary

breakup. In particular, Lozano et al. [136] revealed the linear dependence of the frequency of oscillation against the gas velocity and noted the insignificance of the influence of the liquid velocity. Lozano et al. [136] and Carentz [31] have also shown that the reduced oscillation of the global undulation is strongly related to the momentum ratio gas/liquid. Taking into account the work of Siegler et al. [232] on the influence of the sheet thickness, the flapping of the sheet has been characterised by two non-dimensional numbers: the momentum flux ratio and the reduced frequency of the oscillation. Using the subscripts *gas* and *liq* to refer to the gas and liquid phases respectively, the momentum flux ratio M reads:

$$M = \frac{\rho_{gas} U_{gas}^2}{\rho_{liq} U_{liq}^2} \quad (2.2)$$

The effect of the gas boundary layer on the onset of the sinuous instability has been studied through both stability analysis and experimental approach. Raynal [195], Lozano et al. [135, 136] and Marmottant and Villermaux [145, 146] have concluded that the thickness of the viscous layer — originating from the friction on the injector walls — has a major influence on the scales of instability. More precisely, these studies suggest that the thickening of the boundary layer reduces the growth rate and the wave number for which the growth rate is maximum.

This literature review has highlighted the complexity of the atomisation process. It has also pointed out that the scientific community has a relatively limited understanding of the phenomenon.

As the experimental approach struggles to provide useful insight in the physical mechanisms controlling the primary breakup, research studies suggest that a significant breakthrough could be achieved with the numerical approach.

This chapter identifies and describes the fundamental features of the primary breakup. These will be critical in the choice of a suitable numerical method to simulate the phenomenon.

Chapter 3

Numerical modelling of multiphase flows

The numerical tool provides a framework to calculate the solutions of the Navier-Stokes equations in the idealised representation of an experimental setup. Therefore, it constitutes an intermediate step between the analytical and the experimental approaches. The realism of these numerical experiments can be raised up to a certain limit, depending on the complexity of the problem. More precisely, the level of idealisation concerns principally: the dimensionality of the domain (2D vs. 3D), the complexity of the physical models embedded in the equations (e.g. description of the interface for multiphase flows), the level of sophistication of the boundary conditions (e.g. velocity profiles) and the treatment of the turbulence.

After detailing the mathematical formulation of the problem in Section 3.1, this chapter presents the numerical methods available to capture the physics of two-phase flows. In particular, the modelling strategies handling the smallest scales of the flow field are considered in Section 3.2, the interface description techniques are reviewed in Section 3.3 and the means of treating the singularities associated with the presence of the interface are presented in Section 3.4.

3.1 Problem formulation

Scardovelli and Zaleski [217] provide the detailed derivation of the governing equations for multiphase flows with interface modelling. This modelling is based on the continuum hypothesis, assuming a sharp interface and accounting for the effect of intermolecular forces at the interface by capillarity. In particular, the framework detailed in this chapter leads to the problem formulation for two incompressible immiscible fluids — in the absence of phase change — with relevance to primary breakup (see [54] for justification).

3.1.1 Navier-Stokes equations

In the Navier-Stokes framework the mass and momentum conservation are respectively given by equations (3.1) and (3.2):

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (3.1)$$

$$\frac{\partial \rho \mathbf{u}}{\partial t} = \rho \mathbf{f} + \nabla \cdot (\mathcal{P} - \rho \mathbf{u} \otimes \mathbf{u}) \quad (3.2)$$

with: \mathbf{f} : the body forces (essentially gravity \mathbf{g})

\mathcal{P} : the stress tensor (molecular rate of transport of momentum), given by:

$$\mathcal{P} = (-p + \lambda \nabla \cdot \mathbf{u}) \mathcal{I} + 2\mu \mathcal{D}$$

where: λ : is the second coefficient of viscosity, which is $\lambda = -\frac{2}{3}\mu$, in the Stokes hypothesis

\mathcal{I} : is the unit tensor

\mathcal{D} : is the rate-of-strain tensor: $\mathcal{D} = \frac{1}{2} \left(\nabla \mathbf{u} + (\nabla \mathbf{u})^T \right)$

3.1.2 Incompressible flows

It is generally assumed that both the liquid and the gas are incompressible in multiphase flow modelling which supposes that the speed of the flow is much lower than the speed of sound ($Ma \ll 1$) everywhere in the domain considered. This hypothesis leads to equations (3.3) and (3.4) for the mass and momentum conservation respectively:

$$\nabla \cdot \mathbf{u} = 0 \quad (3.3)$$

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot (2\mu \mathcal{D}) + \rho \mathbf{f} \quad (3.4)$$

This assumption is generally true for most breakup conditions; however Tanguy [246] has questioned its validity in the case of pockets of gas trapped by ligaments merging back with the liquid core. Besides, various compressible formulations of the Navier-Stokes equations have been derived for multiphase flow modelling (see [154, 160, 190, 212]).

3.1.3 Fluid mechanics with interfaces

The whole multiphase domain can be divided into sub-domains filled with the different phases. In each sub-domain, the Navier-Stokes equations hold, whereas on the interfaces the connection between domains is simply obtained through mass and momentum conservation. This leads to the formulation of jump conditions for quantities discontinuous across the interface.

Material properties Introducing $[\]_{\Gamma}$: the jump condition operator across the interface Γ , the jump in material properties is expressed by:

$$[\rho]_{\Gamma} = \rho_{liq} - \rho_{gas} \quad (3.5)$$

$$[\mu]_{\Gamma} = \mu_{liq} - \mu_{gas} \quad (3.6)$$

Mass conservation Without phase change, the continuity of fluid velocity leads to equations (3.7) and (3.8) in the jump notation:

$$\mathbf{u}_{liq} = \mathbf{u}_{gas} \quad (3.7)$$

$$[\mathbf{u}]_{\Gamma} = 0 \quad (3.8)$$

Introducing $\hat{\mathbf{n}}$: the unit normal to the interface, equation (3.8) implies that the velocity of the interface is the normal velocity, V , and that it does not change across the interface:

$$V = \mathbf{u}_{liq} \cdot \hat{\mathbf{n}} = \mathbf{u}_{gas} \cdot \hat{\mathbf{n}} \quad (3.9)$$

NB: In the case of phase change, the normal velocity is given by $u' = \mathbf{u} \cdot \hat{\mathbf{n}} - V$, in the frame of reference moving with the interface. Applying the Rankine-Hugoniot equation, the conservation of mass leads to: $\rho_{liq}u'_{liq} = \rho_{gas}u'_{gas} = q$, where q is the mass transfer from one phase to the other. Expressed in the general frame of reference, it reads: $\rho_{liq}(\mathbf{u}_{liq} \cdot \hat{\mathbf{n}} - V) = \rho_{gas}(\mathbf{u}_{gas} \cdot \hat{\mathbf{n}} - V) = q$

The continuity of normal velocities across the interface, obtained for $q = 0$, is a consequence of mass conservation whereas the continuity of tangential velocities ($\mathbf{u}_t = \mathbf{u} - (\mathbf{u} \cdot \hat{\mathbf{n}})\hat{\mathbf{n}}$) is an assumption.

Momentum conservation The momentum conservation across the interface can be written as the following jump condition:

$$[-p\mathcal{I} + 2\mu\mathcal{D}]_{\Gamma} \cdot \hat{\mathbf{n}} = \sigma\kappa\hat{\mathbf{n}} + \nabla_{\Gamma}\sigma \quad (3.10)$$

which can be split into normal and tangential stress conditions (noting : $\hat{\mathbf{t}}^{(k)}$ two independent vectors tangent to Γ):

$$[-p + 2\mu\hat{\mathbf{n}} \cdot \mathcal{D} \cdot \hat{\mathbf{n}}]_{\Gamma} = \sigma\kappa \quad (3.11)$$

$$[\mu \hat{\mathbf{t}}^{(k)} \cdot \mathcal{D} \cdot \hat{\mathbf{n}}]_{\Gamma} = \hat{\mathbf{t}}^{(k)} \cdot \nabla_{\Gamma} \sigma \quad (3.12)$$

The surface tension term acts as a force concentrated on the interface. The following notations have been introduced:

- ∇_{Γ} the gradient operator restricted to the surface.
- σ is the surface tension. In general, σ is assumed constant along the interface ($\nabla_{\Gamma} \sigma = 0$) which results in a continuous tangential stress across the interface:

$$[\mu \hat{\mathbf{t}}^{(k)} \cdot \mathcal{D} \cdot \hat{\mathbf{n}}]_{\Gamma} = 0$$
- κ is the curvature of the interface, defined by:

$$\kappa = -\nabla_{\Gamma} \cdot \hat{\mathbf{n}} \quad (3.13)$$

3.1.4 Whole domain formulation

A whole domain formulation can be developed as an alternative to the “jump condition form” of the previous section. In this formulation the Navier-Stokes equations are expressed over the whole multiphase domain and the interface is accounted for through singular source terms. These singular terms balance each other thanks to the jump conditions.

The presence of the interface does not involve source terms for the mass conservation. Consequently, the continuity equation remains unchanged:

$$\nabla \cdot \mathbf{u} = 0 \quad (3.14)$$

The momentum balance is given by:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \nabla \cdot (2\mu \mathcal{D}) + \rho \mathbf{f} + \mathbf{f}_{cap} \quad (3.15)$$

The source term $\mathbf{f}_{cap} = \sigma \kappa \delta_{\Gamma} \hat{\mathbf{n}} + (\nabla_{\Gamma} \sigma) \delta_{\Gamma}$ (where δ_{Γ} denotes a distribution concentrated on the interface) has been added to represent the singular capillary forces.

The jump condition on stress (3.10) allows the balancing of the singularities potentially appearing in the terms: ∇p , $\nabla \cdot (2\mu\mathcal{D})$ and \mathbf{f}_{cap} of the momentum equation.

3.1.5 Conservative form

The conservative formulation of the Navier-Stokes equations requires the expression of the capillary effects by a tensor:

$$\mathcal{T}_\sigma = -\sigma (\mathcal{I} - \hat{\mathbf{n}} \otimes \hat{\mathbf{n}}) \delta_\Gamma \quad (3.16)$$

Lafaurie et al. [119] showed that the capillary forces could be written as: $\sigma\kappa\delta_\Gamma\hat{\mathbf{n}} + (\nabla_\Gamma\sigma)\delta_\Gamma = -\nabla \cdot \mathcal{T}_\sigma$. In the absence of body force, this leads to the following conservative form of the momentum balance:

$$\frac{\partial \rho \mathbf{u}}{\partial t} = -\nabla \cdot (p\mathcal{I} + \rho \mathbf{u} \otimes \mathbf{u} + \mathcal{T}_\sigma - 2\mu\mathcal{D}) \quad (3.17)$$

3.2 Numerical framework

The numerical frameworks suitable to model the primary breakup involve interface description techniques. They can be split into two main types: direct numerical simulation (DNS) and large eddy simulation (LES).

3.2.1 Direct numerical simulation of atomisation

The challenges of modelling atomisation

One of the main difficulties of modelling the whole atomisation process comes from the breadth of length and time scales involved in the governing physical phenomena. This requirement is particularly challenging for DNS studies aiming to reduce the uncertainty in the physics by resolving all the length and time scales of the flow. In addition to the Kolmogorov length scale η — the smallest scale in turbulent flows — multiphase flows require the resolution of the smallest liquid structure: ζ . However,

with the change in the topology of the interface, this length scale tends to zero and therefore cannot be resolved using DNS and a certain level of modelling has to be introduced to handle this issue. With these models, a proper simulation of the atomisation process would associate the smallest liquid structure ζ , to the size of the smallest droplets produced and, according to Gorokhovski and Herrmann [71], ζ should be resolved by 2 to 5 grid cells.

The interface thickness — typically a few tenths of nanometres — is orders of magnitude smaller than the smallest scales captured by the DNS (η , ζ). Therefore, the primary breakup, like any other multiphase flow problem, involves a discontinuity in material properties across the interface. Most simulations assume constant density and viscosity in each phase and use a Heaviside function to represent these quantities in the whole domain. In the case of finite volume schemes, the jump in material properties is treated by defining the discrete control volume value of a quantity to be the control volume average. This is the preferred approach for volume of fluid (VOF) and conservative level set approaches [163] as the volume fraction is already available for these methods. For finite difference methods, the evolution of the material properties is described by a smeared out version of the Heaviside function. Finally, the ghost fluid method (GFM) of Fedkiw et al. [55] avoids any smearing of the interface by solving the two phases separately and imposing the jump conditions at the interface. This method requires precise knowledge of the interface location and is therefore geared towards level set methods.

Even at high “global” We , the capillary effects dominate at small scales and their role in the breakup process needs to be captured [146]. In addition, surface tension forces are only active on the (nearly) infinitely thin interface and are therefore singular. It results that another major complexity of modelling the primary breakup — and more generally any multiphase flow — is the numerical handling of surface tension. In order to tackle this issue, Brackbill et al. proposed the continuum surface force (CSF) method [26] which relies on applying the surface tension as a volume force on a transition region of finite thickness which represents the interface

between the phases. Besides the jump in material properties, the GFM [55] also allows for the application of surface tension forces with a pressure jump condition at the interface. The latter approach avoids spreading the capillary forces and maintains a sharp interface. However, neither method guarantees the exact balance between the pressure gradient across the interface and the surface tension forces, with any interface description technique. This leads to the production of “spurious currents”: unphysical vortices in the vicinity of the interface.

The turbulence is believed to play a significant role in the atomisation process both on the primary breakup [54] and on the secondary breakup [7] and [65]. At the typical conditions of injection in aero-engines the Re is high enough so that the 3D effects on the primary breakup become significant and 2D simulations of the atomisation are not relevant. However, depending on the configuration studied, 2D simulations may give some insight into the amplification of instabilities on the interface before the occurrence of non-linearity [65].

The atomisation process involves, per se, very frequent topology changes with the merging and the breakup of the interface surfaces. Depending on the technique employed to describe the interface, some “pinching models” may be required to handle this computational complexity. Implicit interface description methods, such as VOF and level set, handles the topology changes automatically when two interface segments enter a computational cell. Although straightforward for these methods, the topology changes are grid dependent. However, if the grid size is adequately chosen, the smallest droplet produced is significantly larger than the minimum cell size and the grid dependency of the droplet size distribution should be negligible. Explicit interface description methods require the use of numerically expensive algorithms for the breakup or the merging of interfaces but they give the opportunity to introduce physically based models for the pinching.

Besides, the Courant-Friedrichs-Lewy (CFL) condition must account for the additional constraints of the multiphase physics. The restriction on the time step must therefore account for the effect of gravity, viscosity and surface tension [26, 244]. For

time-dependent flows in a compressible framework, Popinet [182] also noted that the speed of propagation of sound waves imposes strong restrictions on the time step. Even though filtering techniques can free the calculation from these restrictions, Popinet's studies suggest that the incompressible formulation is more efficient. Besides, since most primary atomisation applications involve low Ma , solving the NS equation in the variable density incompressible limit can be perceived as a valid assumption and is the approach chosen in most DNS codes.

Despite all these challenges, some successful DNS of atomisation have been obtained with various interface description techniques. On the Diesel jet configuration, it is worth mentioning the simulations of Menard et al. [151, 152] with a novel coupled level set-VOF technique (CLSVOF) and Desjardins et al. [47] applying the conservative level set (CLS) of Olsson and Kreiss [163]. Kim et al. [111] reproduced the round jet in co-flow of the experiments of Marmottant and Villermaux [146] by combining the refined level set grid (RLSG) method of Herrmann [85] with the Lagrangian tracking of the small spherical droplets. The results of these simulations are further detailed in Section 5.1.

Alternatives to full DNS

DNS is limited on both the maximum We and Re achievable in a simulation. More precisely, increasing either the We or the Re generally leads to a more complex flow field with the production of smaller droplets and, in the case of a higher Re , the generation of smaller vortices. It follows that the resolution required to achieve a reasonable accuracy is unaffordable for the simulation of the atomisation process in an aero-engine injector. Consequently, some methods have been developed in order to work around the difficulty of running brutally DNS on this problem.

In parallel to DNS, some phenomenological models have been proposed in order to maintain affordable CPU requirements. The existing models are based on the injection of round blobs from the nozzle and therefore rely on crude simplifications

to represent the primary breakup. However, these models include some of the fundamental mechanisms involved in the breakup process such as interface instabilities [15], jet turbulence [96], cavitation [113] and droplet shedding [278].

The study of atomisers requires modelling upstream of the nozzle (injection channels, etc) and dealing with a broad range of scales: from the large scale instabilities of the jet core to the small scale ligaments. For this purpose, adaptive mesh refinement (AMR) techniques have been developed to scale the simulation with respect to the relevant quantities (regions of large gradients, large vorticity and small curvature of the interface). These methods allow for accurate simulation of the liquid atomisation whilst embedding the complex geometry of the injector. Three different strategies have been followed for the production of efficiently parallelisable AMR techniques:

Patch refinement: this approach has been taken by Sussman et al. [241] for their level set technique. This method involves a hierarchy of nested rectangular grids, in which each level of refinement is represented by the union of the grid patches of a given resolution. In order to avoid any numerical artefact of the AMR, the successive levels of refinement are strictly contained within each other. Sussman et al. focussed the refinement in the vicinity of the interface and on regions of high curvature.

Tree-based adaptive solver: this method was developed by Popinet [182, 183] and combined with a VOF technique. For this technique, the refinement occurs by the equal split of the cell in each of its dimensions, resulting in 4 (respectively 8) leaf cells in 2D (respectively 3D). This type of structure — called quadtree in 2D and octree in 3D — handles conveniently boundaries and interfacial flows and is particularly suited for a multigrid implementation. Popinet combined this technique with the VOF [73] and the marker [185] interface description techniques.

Refined level set grid: this technique is due to Hermann [85, 86]. With this

technique, the solution of the level set equations are solved on a separate, high resolution Cartesian grid. Herrmann [85] combined this method with the “narrow-band” methodology of Peng et al. [173]: the number of multiphase equations solved in a computational cell increase with its proximity to the interface.

Moreover, Herrmann [84] has developed a numerical approach that models the interface-turbulence coupling by concentrating all the vorticity at the interface which, therefore, constitutes a vortex sheet of varying strength. The location and motion of the interface is described using a level set equation which is combined with a Lagrangian formulation to derive an Eulerian transport equation for the vortex sheet strength. In the resulting simplified system of equations, the relevant physical processes occurring at the interface are explicitly described by additional source terms accounting for the surface tension effects and the stretching of the interface. Presupposing that the level-set/vortex sheet method allows studying directly the physical mechanisms occurring at the interface, Herrmann [84] anticipates to use this framework to develop closure models for LES of turbulent primary breakup. However, this approach neglects the viscous boundary layer adjacent to the phase interface which, according to the experimental studies of Marmottant and Villermaux [146], strongly affect the onset of the breakup.

Finally, LES has been used as a practical alternative to DNS in the simulation of primary breakup. Most of the simulations of atomisation presented in the literature are actually “under-resolved DNS”. In particular, the simulations of Diesel jet atomisation run by Bianchi et al. [18, 19] or Villiers et al. [44] (see Section 5.1) are based on an LES formulation for a single phase flow and therefore neglect the interface sub-grid terms. This kind of approach requires grid convergence studies in order to demonstrate that the non-resolved small scale interfacial terms have no impact on the solution.

3.2.2 Large eddy simulation

Application to multiphase flow simulation

The geometry of the injection device (size of injection chamber, turbulent structures developing in the injection channels, wakes behind the separator plates, large turbulent structures in the combustion chamber) has a significant influence in the atomisation process and needs to be embedded in the simulation [44, 65]. This point is further corroborated by the LES simulations of Apte et al. [6, 7] and Riber et al. [200], involving particle laden flows with secondary breakup models. Their results — validated against experimental data — highlight the importance of large scale turbulence and secondary breakup in the resulting droplet size and velocity distributions. The consideration of such a significant amount of geometry suggests the use of LES which, according to [44]: “only resolves the eddies large enough to contain information about the problem geometry”.

The application of LES to the simulation of atomisation implies the description of the physical mechanisms occurring at the non-resolved scales by ad hoc models that predict small scale behaviour based on large scale parameters. Existing secondary breakup models [167, 168, 196, 197, 248] assume simple geometries for the liquid structures in order to simplify the description of their interaction with the gas phase. However, this simplification does not hold in the case of the primary breakup which has principally been studied numerically via the full description of the interface dynamics. The adoption of an LES methodology that makes this approach computationally affordable, is based on the ensemble averaging or the spatial filtering of the NS equations and the production of closure models for the sub-grid terms [28, 29]. However, this task is challenging and besides the effect of surface tension, very little work has been done to account for the physical mechanisms occurring at the interface.

In two-phase flows, turbulent energy can be exchanged between phases through the interface. Experimental [120] and numerical [62] studies highlight the importance

of the interaction between the turbulence and the interface. In particular, DNS studies of Fulgosi et al. [62] on the influence of the multiphase topology on the turbulence showed that a moving interface does not behave as a solid wall: for a moving interface the level of anisotropy is lower and the turbulence transfer is increased.

Based on these findings, Labourasse et al. [118] stress the importance of accounting for the complex turbulence-interface interactions in LES of two-phase flows. In particular, the authors highlight the need to produce sub-grid models of the non-linear transfer of turbulence energy across the interface. In addition, the authors noted that, in the transfer of turbulent energy between phases, the oscillations of the interface lead to the modification of the turbulence energy spectrum: absorption of high frequencies and release of lower frequencies.

Sub-grid scales

Labourasse et al. [118] have conducted an analysis of the turbulence-interface coupling using DNS (VOF and front-tracking methods) in order to study the sub-grid transfer through the interface. This work focused on the theoretical basis and the development of the LES formulation (spatial filtering, modelling of the non-resolved scale contributions) for multiphase flows.

Labourasse et al. identified three categories of closure terms for the LES of multiphase flows:

The “classical” sub-grid terms: they reproduce the effect of “sub-grid scales correlations on the resolved flow”. These terms are common with the single phase studies of turbulent flows and, according to [118], a large amount of the existing models (reviewed in [211]) can be re-used for LES of multiphase flows. However, the relative magnitude of these terms may differ from the classification of Vreman [267, 268] in the vicinity of the interface.

The “pure interfacial” sub-grid terms: they represent the “sub-grid contribu-

tion of the interface characteristics” and can be replaced by a jump condition at the interface.

The misrepresentation of the “mixed phase volume of fluid”: the cells surrounding the interface contain a mixed phase; their representation by thermodynamical and physical laws leads to a sub-grid error. The study of these terms was tackled by Duquennoy et al. [51] and Mathieu et al. [149] in their work on contact lines, but no models are currently available for these effects.

Labourasse et al. [118] discussed the relative magnitude of the various sub-grid scale terms and the precision of the different filtering techniques. In particular, the authors noted the better performance of mass-weighted filtering processes and recommended it for LES of turbulent two-phase flows. In their discussion on the sub-grid terms, Labourasse et al. reported that the most important ones are related to inertia as in single-phase flows. However, the modification of these sub-grid effects near the interface excludes the use of a viscosity assumption to model them. The correct behaviour for these terms has been reproduced by Boivin et al. [21] with a “Bardina-like-Smagorinsky-like” model. Besides, in their study, Labourasse et al. also proposed a model for the capillary-induced term.

3.3 Interface description

3.3.1 Overview

The most popular numerical methods accounting for the presence of the interface in the computational domain transport the interface either explicitly or implicitly. More recently hybrid methods have been produced to combine the advantages of several approaches at the expense of CPU time. A fourth group of methods gathering more unusual techniques such as smoothed particle hydrodynamics (SPH), is also reviewed in this sub-section under the label “alternative methods”.

Explicit interface description

The changes of topology have to be specified for these methods and it gives the opportunity to implement some physics-based breakup models. In return, explicit interface description techniques require the introduction of cumbersome algorithms to handle merging and breakup of the interface surfaces.

Moving mesh This approach is based on the Lagrangian transport (according to the underlying velocity field) of the grid nodes associated with the interface location. This technique is particularly geared towards the simulation of small amplitude motions of the interface and has given accurate results in the case of small waves and weakly deformed bubbles. In particular, this method has been applied successfully by Ryskin and Leal [209, 210] to study rising bubbles on a curvilinear grid and by Magnaudet et al. [138] in the calculation of the flow around spherical bubbles.

As reported by Hirt et al. [88], in the case of pure Lagrangian transport of the interface nodes, the mesh can become highly distorted even if frequent regridding and rezoning are applied during the computation. In order to mitigate this issue, Hirt developed an arbitrary-Lagrangian-Eulerian (ALE) method [87] for the transport of the grid points. In addition, depending on the flow considered, it can also be necessary to add or remove grid points. This can be done using the specific algorithms presented in [61] and [66].

Front tracking The front-tracking method combines the resolution of the flow properties on a fixed grid with the Lagrangian transport (according to the underlying velocity field) of a web of massless particles — typically a triangular mesh in 3D — representing the phase interface. The capillary forces are represented by the CSF method and interpolated from the interface web to the fixed grid. In line with the CSF approach, the discontinuities across the interface are smoothed by the use of delta functions in the transition region between the phases. This technique has

been used principally by Glimm et al. [69] and Univerdi and Tryggvason [261]. In particular, the latter authors extended it to 3D [260] and applied it to the simulation of primary breakup [259].

The main advantage of the front-tracking method is the lack of mesh dependency of the breakup process. In addition, despite not being inherently mass-conservative, front-tracking methods generally conserve mass with a very high level of accuracy. Also, the use of high-order polynomial interpolations in conjunction with surface tracking can lead to very accurate calculations of surface tension [184, 185].

Surface markers allow for simulation of very thin liquid bridges with high accuracy; however Scardovelli and Zaleski [217] noted that this is only an advantage in the absence of surface tension and when the phases have the same densities and viscosities: “interface transparent to the fluid”. Otherwise, the sub-grid scales of the velocity and pressure fields will not be resolved without a modification of the fixed grid.

Moreover, this technique requires the introduction of heavy algorithms to handle the changes of interface topology. Also, the topological changes and the interface distortion can lead to the concentration or the stretching of the marker distribution. There is, therefore, a need to redistribute these particles and re-mesh the interface. All of these processes need to be done efficiently in the case of the atomisation where the breakup and merging of the interface occurs frequently.

Implicit interface description

Implicit interface description methods — often referred to as front capturing — represent the interface through a scalar field advected in a fixed grid. The extension to 3D formulations of front-capturing techniques is therefore straightforward. In addition, with such techniques, the changes of the interface topology are handled automatically as soon as two interface segments enter the same computational cell.

This feature of front-capturing techniques leads to both the inaccurate predic-

tion of the breakup time and the undesirable grid dependency of the pinch off and reconnection processes. However, Gorokhovski and Herrmann argue that if the grid is fine enough to capture the fundamental mechanisms of the breakup, the solution should remain acceptable [71]. Consequently, in order to carry any physical significance, the front-capturing simulations of the primary breakup have to show grid independence.

The volume of fluid and level set approaches are the two main implicit interface description methods. Both techniques are most relevant to the modelling of primary breakup and a whole section has been dedicated to each method. The phase field method (PFM) — also describing the interface implicitly on a fixed Eulerian grid — is briefly outlined below.

Phase field method This numerical method describes two-phase flows from the Van der Waals-Cahn-Hilliard theory of interfaces. This theory considers the fluid at a mesoscopic scale intermediary between the macroscopic scale of continuum mechanics and the molecular scale. It leads to a modelled interface of finite thickness across which the velocity and density vary continuously. The formulation of the theory is based on Van der Waals model [264] for the free energy density, f , (3.18), in which C is the “composition variable” and Ψ is C ’s “chemical potential”.

$$f = \frac{1}{2}\alpha |\nabla C|^2 + \beta\Psi(C) \quad (3.18)$$

Cahn and Hilliard extended this formulation to time-dependent flows and produced a modified version of the Navier-Stokes equations embedding a “continuum forcing” term [30]. In this framework, the surface tension is derived from the energy density field.

PFM transports the interface using standard advection techniques and is, therefore, easy to implement in 3D, even on unstructured grid. Also, because of its potential for energy dissipation the numerical implementations of this method are

generally free of parasitic flows. However, the structure of the modelled interface is crucial to the calculation of the surface tension and the numerical implementation of PFM generally lead to thick interfaces (typically 8 cells wide). In order to address this major drawback, Jacqmin proposed a “compact method” [101] that thins up the interface (2 cells wide) at equal accuracy. However, Jacqmin acknowledges the difficulty of assessing a simulation’s accuracy and noted the sub-linear formal convergence rate of the method ($O(h^{4/5})$; “higher in practice”).

Jacqmin was one of the first to use such methods to model two incompressible phases [100]. PFM applications concern in particular the simulation of moving contact lines [102, 223] and interface topology changes [134].

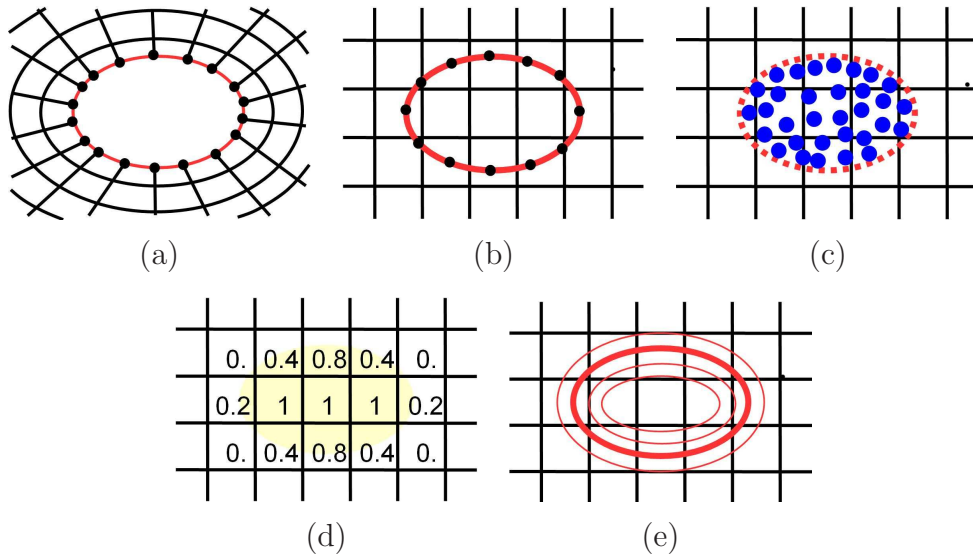


Figure 3.1: Main interface description methods: (a) Moving mesh; (b) Front tracking; (c) Marker and cell; (d) VOF; (e) Level set

Hybrid methods

Some hybrid methods have been developed: such as marker-LS, marker-VOF, LS-VOF, in order to combine the advantages of the different approaches. The motiva-

tion for coupling methods with LS is the advantage of introducing a smooth function yielding the second-order derivatives needed to calculate the surface tension.

From the point of view of the LS, hybrids methods have been proposed to make use of better volume conservation properties of auxiliary interface-tracking methods to correct the LS representation of the interface. In particular, the LS has been coupled with marker particles — to produce the particle LS (PLS) of Enright et al. [53] — and with the VOF method that lead to two notable methods: the coupled LS and VOF (CLSVOF) of Bourlioux [25] (popularised by Sussman and Puckett in 2000 [243]) and the mass-conserving level set method (MCLS) of Van der Pijl et al. [263]. Both techniques achieve the coupling by defining the volume fraction C , from the level set function ϕ as in equation (3.19). Introducing H , the Heaviside function and E_i , the grid cell considered (3.19) reads:

$$C(E_i) = \frac{1}{|E_i|} \int_{E_i} H(\phi) \, dE_i \quad (3.19)$$

Hybrids of the VOF and marker particle methods have been produced in order to improve the interface reconstruction, limit spurious currents and introduce physical models for the changes in interface topology. In particular, Lopez et al. produced a 2D marker-VOF method [133] to reduce the flotsam (isolated blobs of liquid, artefact of the VOF interface reconstruction algorithm) and track liquid structures thinner than the mesh size. The technique adds to the VOF approach some marker particles at the mid-point of every cell interface segments and reconstructs the interface using cubic-splines interpolations. Although the addition of marker particles results in a better control of the interfacial topology changes, the extension of this method to 3D is very difficult.

Alternative methods

Boundary integral (BI) The boundary integral formulation [132] is based on three assumptions: the viscous forces are negligible compare to the surface tension

forces (inviscid flow and therefore Euler equations), the density of the liquid is much higher than that of the gas (incompressible liquid in a vacuum) and the flow — initially irrotational remains irrotational. Under these assumptions, the interface reduces to a free boundary and can be associated to a vortex sheet [108]. The BI method simulates the evolution of this vortex sheet by solving an integral equation along the interface. Despite its severe limitations to simple multiphase flows, this technique has the advantage to concentrate the mesh on the interface. This method has been applied to jet breakup [143] and pendant drop [219].

Marker and cell (MAC) The oldest computational methods for tracking an interface is the “marker and cell” approach of Harlow and Welch [76] formulated in 1965 for free-surface flows where the motion of the light phase is neglected. This method use both a fixed grid — to simulate the motion of the dense phase in the region delimited by the free-surface — and massless “volume markers” homogeneously distributed in the dense phase and transported with a Lagrangian scheme. This method, extended to two-phase flows by Daly [40], handles topology changes automatically and conserves mass reasonably well. However, the MAC approach requires a large number of markers per grid point in order to capture the interface accurately and avoid high levels of numerical diffusion. Besides, as for surface marker methods, the flow field may concentrate the particles in places and some specific algorithms have to be implemented to redistribute the markers.

Lattice Boltzmann method (LBM) The LBM (reviewed in [204, 205]) associates a continuous fluid to a granular fluid, assuming that the motion of both types of fluid are similar on a large scale. Instead of solving the Navier-Stokes equations, this method describe the motion of the particles composing the fluid by solving the discretised Boltzmann equation combined with a collision model for the particles. This method emerged from the lattice-gas automata (LGA, [17]) which is based on a simplified model of the particle dynamics. In the LGA model, the

particles can only move between the nodes of a hexagonal lattice and there can only be 0 or 1 particle at a certain node moving in a given “lattice direction”. The main disadvantage of this simple model is the potential rise of statistical noise due to the Boolean description of the number of particles travelling in a lattice direction. The LBM solved this problem by using ensemble averages instead of Boolean numbers. A major drawback common to both LGA and LBM is that these methods describe the fluid statistically and not necessarily physically.

Smoothed particle hydrodynamics (SPH) Originally developed to solve problems in astrophysics [68], SPH represents the fluid by a set of particles of properties distributed over a “smoothing distance” — varying in space and time — using a kernel function. At a given point, the properties of the fluid are then obtained by summing the relevant quantity over all the particles in the range of the kernel, so that the fluid is simulated by the collective properties of the particles. The motion of the particles is driven by external forces and their mutual interaction. Depending on their relative distance the particles can either attract (long range) or repel (short range) each other [155, 156]. The surface tension is then obtained by the repulsion forces between particles belonging to different phases. With SPH, a large number of phases can be simulated by introducing as many types of particle and the changes in interface topology are handled automatically. However, this technique is computationally intensive and is less attractive than implicit interface description methods (VOF and LS) for simple two-phase flow situations.

3.3.2 Formulation of the VOF method

This method captures the interface through the transport of the volume fraction, $C_{i,j}$, in the computational domain. The volume fraction — also known as colour function — represents the volume (respectively the area) occupied by one phase in a 3D (respectively 2D) computational cell. Consequently, the colour function takes

values between 0 and 1 in the cells cut by the interface and either 0 or 1, depending on the phase, in the rest of the domain (see consistency equation (3.20)). However, numerical errors can drive the volume fraction out of this range and some specific algorithms have to be implemented to avoid non-physical values of $C_{i,j}$.

$$0 \leq C_{i,j}^m \leq 1 \quad (3.20)$$

In 1974, DeBar implemented the first VOF method to simulate multiphase flows with a 2D Eulerian formulation for compressible flows [45]. Nowadays, most implementations of the VOF method are based on an incompressible formulation of the Navier-Stokes equations because of the resulting level of simplification. However, some authors have implemented the VOF technique in the compressible formulation: see for example [154, 160, 190].

The VOF method involves two stages: the reconstruction of the interface followed by the transport of the volume fraction. Both of these steps only require the knowledge of the volume fraction and the velocity field (for the latter algorithm) in the cell considered and its neighbours. This makes the VOF method local and therefore highly parallelisable. Also, introducing d , the dimension of the computational domain and N , the number of cells in one dimension, the advection of the interface scales as $O(N^{d-1})$ which is small compare to the update of the velocity field.

The major advantage of the VOF method is that it is inherently mass-conservative (expressed for VOF methods by equation (3.21)). More precisely, in an incompressible formulation, if the technique is implemented with the appropriate numerics (conservative numerical schemes) it conserves exactly the mass of each phase.

$$\sum_{i,j} C_{i,j}^{m+1} = \sum_{i,j} C_{i,j}^m \quad (3.21)$$

The main drawbacks of the VOF method are the uncertainty on the curvature calculation (approximate interface reconstruction) and the diffusion of the interface

(CSF). In the case of under-resolution (radius of curvature of the order of the grid size), this leads to inaccuracies in the calculation of the capillary forces and the frequent occurrence in VOF simulations of spurious currents (see Section 3.4). However, the method remains robust in the limit of very large curvature of the interface typically encountered during topology changes.

Interface reconstruction algorithm

The determination of the interface curvature — needed for the calculation of the surface tension — requires the knowledge of the interface shape. For this purpose the “interface reconstruction” algorithm is implemented in the method and, based on the volume fractions in a 3×3 block in 2D ($3 \times 3 \times 3$ block in 3D), provides the interface location in the cell at the centre of the block.

A broad variety of methods have been developed over the years which can be gathered in three main groups of algorithms: the simple line interface calculation (SLIC), the piecewise linear interface calculation (PLIC) and the piecewise parabolic interface calculation. In the case of a linear interface reconstruction, the various methods differ on the means to determine the normal $\hat{\mathbf{n}}$ to the segment. Once the orientation of the surface is known, the approximated interface is completely defined by the volume fraction in the cell considered.

Novel interface reconstruction methods have been developed in order to reduce the discontinuities at the cell faces caused by the PLIC representation. In particular, Manservigi et al. have proposed a method that does this in two steps for a 2D computational domain [139]. At first, the average curvature is calculated at the cell considered using a PLIC algorithm with a least square procedure in a 3×3 block. Then, from the two intersections of the fitted circle with the face of the cells, an internal point is added to better approximate the curvature and conserve the colour function.

Simple line interface calculation These methods reconstruct the interface in each cell by a segment aligned with the grid. Being very simple, SLIC algorithms [89, 159] are usually crude (at best first-order accurate) and lead to the production of a large amount of flotsam in the transport of the colour function, even in the case of simple velocity fields (translation, rotation). The original method of Noh and Woodward [159], calculating the segment location from a 3×1 block, has been improved by Chorin [35] — who reconstructs a “stairs-shape interface” by using a 3×3 block — and later by Lafaurie et al. [119]. Because of the geometrical specificities of an interface reconstructed using SLIC, this method is coupled with the split advection algorithm.

Piecewise linear interface calculation Amongst the various algorithms produced for approximating the interface, the most commonly used is the piecewise linear interface construction (PLIC) method which reconstructs the interface in each cut cell by a line in 2D (a plane in 3D). Most PLIC algorithms reconstruct the interface as a discontinuous sequence of segments in 2D (polygons in 3D) with “asymptotically small discontinuities” [217]. The generation of a continuous interface is generally perceived as an unnecessary complexity.

Already implemented in the earliest VOF method of DeBar [45], the PLIC method continues to attract the interest of many authors. In their review of the VOF algorithms, Pilliod and Puckett assessed the performance of the main PLIC interface reconstruction schemes against their convergence rates with grid refinement [178]. The test problems used by Pilliod and Puckett involved the typical approximation of lines, circles, crosses and slotted disks [284].

Based on the principle that a second-order PLIC algorithm reconstructs exactly a linear interface, Pilliod and Puckett categorise the methods of Puckett and Saltzman [190] (centre of mass), Parker and Youngs [172] and the central difference algorithm as first-order accurate whereas the least square volume of fluid interface reconstruction algorithm (LVIRA) [188] and the efficient LVIRA (ELVIRA) [176] demonstrate

second-order accuracy. Notable VOF-PLIC algorithms have also been produced by Ashgriz and Poo [9], Li [129], Margolin et al. [144] and Rider and Kothe [202].

Piecewise parabolic interface calculation In order to improve the accuracy of the curvature estimation and therefore reduce the spurious currents, Renardy and Renardy have proposed an algorithm that approximates the interface by piecewise paraboloids (in 3D) through least square fit [198]: parabolic reconstruction of surface tension (PROST). Their method reduces significantly spurious currents and satisfies spatial convergence but these advantages comes at great computational expenses.

Propagation of the interface

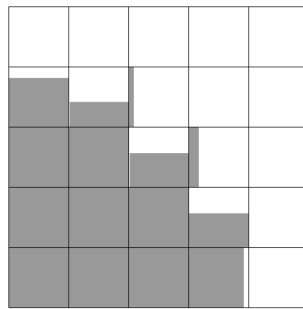
Following the reconstruction of the interface, the volume fraction is transported by the velocity field through advection algorithms. The volume fraction field is then updated at the new time step according to the reconstructed interface and the velocity field at the current time step. The calculation of the fluxes at the cell boundaries is done either independently along each coordinate direction with an “operator split” advection algorithm [129, 218, 238] or with an “unsplit” (or “multidimensional”) algorithm [36, 177, 202] that transports the volume fraction in one step. Both split and unsplit advection schemes are generally based on a conservative formulation of the conservation law for the colour function (equation (3.22)):

$$\frac{\partial C}{\partial t} + \nabla \cdot (C\mathbf{u}) = 0 \quad (3.22)$$

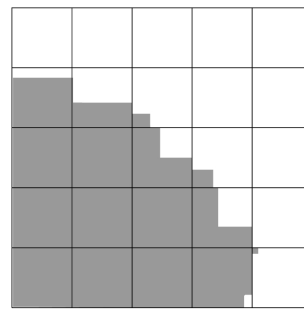
Unsplit algorithms are more complex but lead to more accurate simulations as they account for the fluxes in the diagonal directions (e.g. from (i, j) to $(i + 1, j + 1)$). The accuracy of the advection algorithms is typically assessed by standard tests consisting in the transport of simple geometric shapes such as lines, crosses, circles and slotted disks [284] in a constant velocity field (translation, rotation, reverse single vortex of Rider and Kothe [202]).

| | | | | |
|-----|-----|------|------|------|
| 0 | 0 | 0 | 0 | 0 |
| 0.8 | 0.4 | 0.05 | 0 | 0 |
| 1 | 1 | 0.7 | 0.1 | 0 |
| 1 | 1 | 1 | 0.6 | 0 |
| 1 | 1 | 1 | 0.95 | 0.01 |

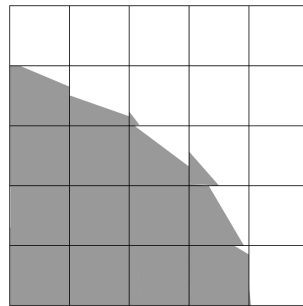
(a)



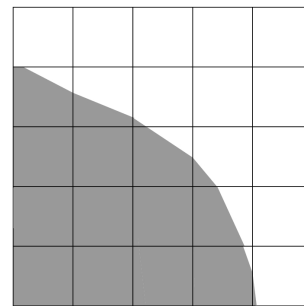
(b)



(c)



(d)



(e)

Figure 3.2: Main interface reconstruction techniques for VOF methods: (a) Ideal interface; (b) SLIC; (c) Chorin (1980)’s technique; (d) PLIC; (e) FLAIR

The advection of the colour function is both critical, since it drives the mass conservation properties of a VOF method, and difficult (complexity of “splitting” the incompressibility condition). In particular, limitations of some existing advection schemes lead to the violation of the consistency condition (3.20) or the random production of “wisp” (different from “flotsam”, see [80]) in the bulk of each phase

($C_{i,j} = \varepsilon$ or $C_{i,j} = 1 - \varepsilon$ with $\varepsilon \ll 1$). Such inconsistencies of advection algorithms have been mitigated by the use of diffusion algorithms to redistribute the volume fraction in neighbouring cells [119] or the filtering of the colour function resulting in the loss of mass conservation (3.21).

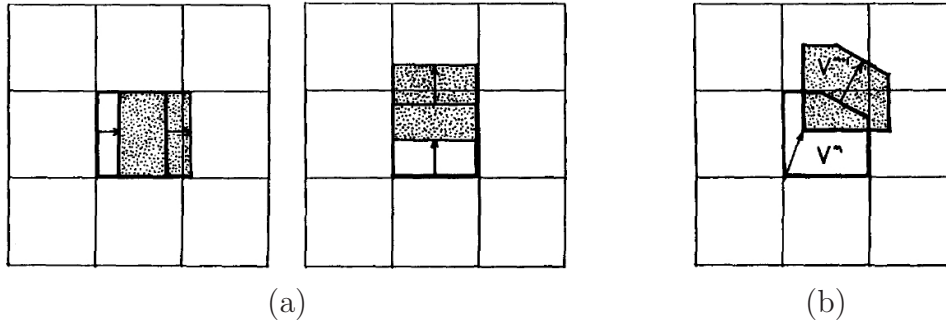


Figure 3.3: Interface advection algorithms for VOF methods: (a) Operator split advection; (b) Unsplit advection

Operator split advection algorithm This method calculates sequentially the fluxes at the cell boundaries in each coordinate direction. Each advection step in a given direction is immediately followed by an interface reconstruction based on the collection of intermediate (or final for the transport in the last direction) volume fractions. The calculation of the fluxes is done by geometric construction given the latest approximated interface in each cell. This approach requires therefore as many interface reconstructions per time step as there are dimensions in the computational domain. By alternating the sweep direction (“Strang splitting”, [238]), the advection algorithm can reach second-order accuracy. Pilliod and Puckett combined the “Strang splitting” with the ELVIRA interface reconstruction algorithm to obtain a formally second-order accurate interface-capturing algorithm [178].

Unsplit algorithm Unsplit algorithms transport the colour function in one step by considering the time evolution of the interface and anticipating the possible

distribution of heavy fluid ($C_{i,j} = 1$) in the cell. Despite requiring only one interface reconstruction per time step, multidimensional schemes are more complex than operator split algorithms (treatment of fluxes in the diagonal directions) and therefore harder to implement.

Both first-order [16, 36] and second-order [10, 80, 81, 177, 202] accurate algorithms have been developed over the years. Unsplit algorithms produce better results than fractional step methods, especially for complicated problems — e.g. unstable displacement in porous media — where split methods (both first and second-order accurate) generally lead to the distortion of the interface (“push-pull” or “staircase” phenomenon, see [16]). In addition, unsplit algorithms resolve better regions of high variation of the derivatives such as corners of the interface.

3.3.3 Formulation of the level set method

The LS formulation is based on the transport of a continuous function, $\phi(\mathbf{x}, t)$, in the computational domain according to the underlying velocity field. In this framework, the ϕ_Γ level set ($\phi_\Gamma = 0$ usually) of the function ϕ represents the interface and ϕ takes values below ϕ_Γ in one fluid and above ϕ_Γ in the other [169, 170, 228]. From a numerical point of view, a smooth function is desirable and ϕ is generally taken as the signed distance function from the interface.

$$|\nabla\phi| = 1 \tag{3.23}$$

The background theory for this method was developed by Sethian [224–227] and the level set technique was originally formulated in [170] to follow fronts propagating with curvature-dependent speeds (crystal growth, flame propagation). For such problems, the equations of motion take the form of Hamilton-Jacobi equations with parabolic right hand sides that account for the curvature effects. This approach has been implemented in a compressible framework by Mulder et al. [157] to produce a compressible level set formulation for two-phase flows.

The algorithms developed by Osher and Sethian [170] to follow the propagation of fronts are based on techniques from hyperbolic conservation laws. The authors rely upon essentially non-oscillatory (ENO, [77, 79, 230, 231]) numerical schemes to cope with the formation of large gradients in front propagating problems and preserve the sharpness of the interface. Such numerics have been further developed by Liu et al. [131] and later by Jiang and Shu [107] who proposed weighted ENO (WENO) schemes that offer better precision/robustness trade-offs.

The level set method operates in three steps: at first, the level set function is defined as the signed distance function (3.23) between any point of the domain and the interface: $\phi = 0$ on the interface, $\phi > 0$ in one phase, $\phi < 0$ in the other phase. Then, the scalar ϕ is transported according to the velocity field (3.24). Once advected, ϕ is no longer a signed distance function and a “re-distancing” algorithm (Section 3.3.3) is applied to maintain ϕ as a signed distance function to the interface and prevent ϕ from becoming irregular.

$$\frac{\partial \phi}{\partial t} + \mathbf{u} \cdot \nabla \phi = 0 \quad (3.24)$$

Like the VOF method, the LS technique handles topology changes naturally. The main advantage of this method is that it provides the precise front location directly (no interface reconstruction). This results in the accurate determination of the interface geometric properties (normal: (3.25) and curvature: (3.26)) and therefore better calculation of surface tension forces. Besides, the LS formulation is easy to implement and can be formulated in N dimensions. In particular, since it does not reconstruct the interface, this method can be extended to 3D more easily than VOF which requires specific algorithms for this purpose.

$$\hat{\mathbf{n}} = \frac{\nabla \phi}{|\nabla \phi|} \Big|_{\phi=0} \quad (3.25)$$

$$\kappa = -\nabla \cdot \left(\frac{\nabla \phi}{|\nabla \phi|} \right) \Big|_{\phi=0} \quad (3.26)$$

The major drawback of this method is that it does not inherently ensure mass conservation. Although the implementation of an iterative re-initialisation algorithm [244] partially corrects this issue, the method does not conserve mass accurately. In order to address this, various methods have emerged: modified re-distancing algorithms [32, 242], refinement strategies [86] coupling with VOF [150–152, 243, 263], coupling with marker particles [53] or alternative definition of the scalar field (conservative LS, [163]).

Re-initialisation algorithm

The advection of the scalar ϕ (3.24) does not ensure that the level set remains a signed distance function and certain multiphase flow features (high velocity gradients, interface topology changes) can lead to an irregular level set function (jump at the interface, spreading). In such instances, the level set function no longer satisfies the condition (3.23) which is yet required for the precise location of the interface and the accurate calculation of the its geometric properties ((3.25) and (3.26)). It is therefore necessary to add a constraint on the transport of ϕ to maintain the condition (3.23) without moving the front. This operation, performed by the re-distancing algorithm, keeps the interface thickness constant.

Originally, re-distancing routines were explicitly searching for the zero level set to re-initialise ϕ around the interface. These techniques were computationally expensive [34] and generally lead to severe distortions of the interface. The re-distancing algorithm of Sussman et al. [244] bypasses the search of the interface by correcting iteratively the position of the level sets — with respect to the zero level set — such that (3.23) is satisfied.

In that method, the constraint added on the advection of ϕ takes the form of a partial differential equation (3.27) solved on an artificial time step at the end of each physical time step. Once the artificial steady state reached in the whole domain,

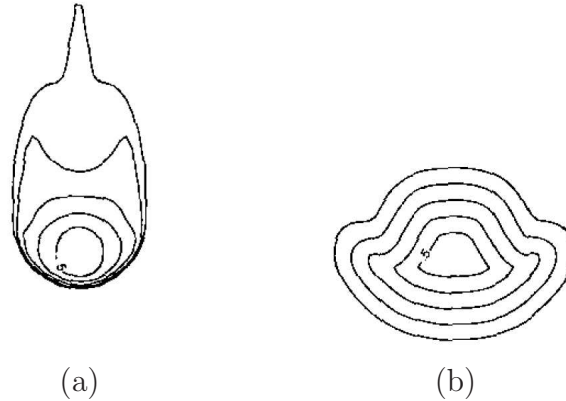


Figure 3.4: Level sets of a water drop falling under gravity [244]: (a) No re-initialisation; (b) Re-initialisation

the re-initialisation is complete. Introducing τ , the artificial time step of the re-initialisation procedure and sign , the sign function mollified around the interface, this equation reads:

$$\frac{\partial d}{\partial \tau} = \text{sign}(\phi) (1 - |\nabla d|) \quad (3.27)$$

$$\text{with: } d(\mathbf{x}, t, \tau = 0) = \phi(\mathbf{x}, t)$$

This algorithm formally conserves the position of the interface. Indeed, when the re-initialisation is complete, either $|\nabla \phi| = 1$ and the property of signed distance function is recovered or $\text{sign}(\phi) = 0$ and \mathbf{x} belongs to the interface. At the end of the iterative process, the level set $\phi(\mathbf{x}, t)$ is reset to $d(\mathbf{x}, t, \tau = \tau_{final})$ in the whole domain.

The accurate calculation of the interface geometric properties requires the level set to satisfy (3.23) only in the vicinity of the interface. Since this algorithm starts the re-initialisation near the interface and propagates outwards in the normal direction, the re-initialisation of the level set is generally obtained with few iterations per physical time steps (2 to 3 in practice).

Despite formally maintaining the position of the zero level set, the algorithm of Sussman et al. still moves the interface slightly in practice. To mitigate this and im-

prove the mass conservation properties of the level set method, Sussman et al. [242] and Chang et al. [32] have proposed modified versions of the re-distancing algorithm based on the addition of a constraint (for instance related to the curvature of the interface) in the solution of (3.27). However, these methods do not systematically re-distribute the mass appropriately (where it was originally lost) and therefore still do not satisfy mass conservation exactly.

Conservative level set

Olsson and Kreiss proposed in 2005 the “conservative level set” (CLS) method [163] based on an alternative definition of level set function. Instead of a signed distance function, the method transports a hyperbolic tangent function $\psi(\mathbf{x}, t)$, that resembles a smeared out liquid volume fraction. As a result, ψ goes smoothly from 0 (in one phase) to 1 (in the other phase) in a transition region of constant thickness. The location of interface is then given by the $\psi_\Gamma = 0.5$ level set. Recalling that ϕ is the signed distance from the interface and introducing ϵ , a parameter controlling the thickness of the interface, ψ is defined by:

$$\psi = \frac{1}{2} \left(\tanh \left(\frac{\phi}{2\epsilon} \right) + 1 \right) \quad (3.28)$$

The level set is transported in two steps: starting with the advection and followed by the re-initialisation of the level set that maintains the thickness of the transition layer constant and preserves the smoothness of ψ 's profile. In the case of an incompressible flow, these equations read:

$$\frac{\partial \psi}{\partial t} + \nabla \cdot (\mathbf{u}\psi) = 0 \quad (3.29)$$

$$\frac{\partial \psi}{\partial \tau} + \nabla \cdot (\psi(1 - \psi) \hat{\mathbf{n}}) = \epsilon \nabla \cdot (\nabla \psi) \quad (3.30)$$

Both steps are implemented with conservative numerical schemes in order to ensure

that ψ is conserved in a divergence free velocity field. However, as the interface has a finite thickness, this property only leads to the conservation of the volume (in 3D) bounded by the $\psi_\Gamma = 0.5$ level set in the case of a flat interface [47]. It results that the volume conservation of the CLS method improves as ϵ decreases for an arbitrary shape of the interface.

However, a minimum resolution of the level set profile is required for the accurate calculation of the interface geometric properties ((3.25) and (3.26)) so that Olsson and Kreiss recommend: $\epsilon = \Delta x/2$, where Δx is the mesh spacing. In the case of the complex velocity field and topology changes — typical of primary breakup situations — no satisfactory trade-off can be obtained for ϵ and the inaccurate calculation of $\hat{\mathbf{n}}$ and κ lead to numerical instabilities. Starting from that method, Desjardins et al. [47] attempted to mitigate this issue by reconstructing the signed distance function ϕ from ψ using Sethian’s fast marching methods [229].

In conclusion, the CLS method reduces the numerical errors accumulated in the advection of the level set function and, as a result, significantly improves the mass conservation for the same computational cost as the standard LS technique. Like hybrid methods, such as PLS [53] or CLSVOF [152, 243, 263], the technique of Olsson and Kreiss partially solves the mass conservation problem of the level set method but at a much lower cost. Therefore, the conservative level set method is a good starting point to develop a numerical method for the simulation of the atomisation process. It is however necessary to improve the numerical behaviour of the method in terms of accuracy and stability.

3.4 Treatment of singularities

After presenting the physical origin of the surface tension, this section gives a brief overview of the principal means of handling singularities numerically. The emphasis has been placed on the descriptions of the continuum surface force (CSF) [26, 175] and the ghost fluid method (GFM) [55] as these two techniques are particularly

popular in multiphase CFD.

3.4.1 Physical origin of the surface tension

Microscopically, the surface tension is due to an imbalance of molecular forces in the liquid, close to the interface. In the bulk of the liquid, the intermolecular forces, maintaining the cohesion of the medium, are at equilibrium as each molecule is pulled equally in all directions by the surrounding molecules (see Figure 3.5). At the interface, an imbalance arises due to the lack of — or the difference in — the attractive force from the molecules constituting the medium beyond the free surface. Consequently, the molecules located at the interface are pulled inwards by the molecules deeper in the same phase. As this inward force is counterbalanced by the resistance of the liquid to compression (short range repulsion forces between the constitutive molecules) equilibrium is reached at a state of minimal energy which corresponds to a minimum of interfacial area.

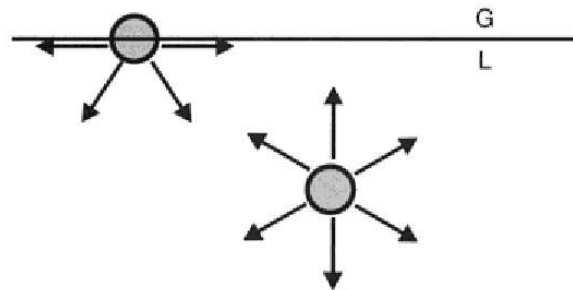


Figure 3.5: Attractive interactions for two molecules: one in the bulk of the liquid and the other one at the interface [42]

In continuum fluid mechanics, it translates as a surface force acting locally — in both normal and tangential direction — on the fluid particles of the interface. The liquid surface then resembles an “elastic skin” in tension and this tension can be perceived as the necessary force to maintain the energetic equilibrium of the

molecular cohesion [42]. Upon certain conditions (presence of surfactant, etc ...), the surface tension may vary along the interface which results in the flow from high surface tension regions to low surface tension regions.

The balance of the normal stresses of two static fluids meeting at an interface of zero thickness is given by the Young-Laplace equation (3.31). This equation relates the pressure difference across the interface to the shape of the interface and the surface tension σ . Introducing the principal radii of curvature R_1 and R_2 at the point of the surface considered, (3.31) reads:

$$\Delta p = \sigma \left(\frac{1}{R_1} + \frac{1}{R_2} \right) = \sigma \kappa \quad (3.31)$$

3.4.2 Continuum surface force

The CSF method is based on the smoothing of the discontinuous quantities over a thickened interface (typically 2 to 3 cells wide). It applies the surface tension as a volume force in this transition region (see Figure 3.6). This technique, generally associated with the whole domain formulation (see Chapter 3.1), involves introducing mollified Heaviside and Dirac delta functions in order to make discontinuous quantities differentiable.

Although robust and easy to implement, this method does not always reproduce faithfully the physics near the interface, as it diffuses the surface location and the associated jump conditions. It also relies upon a choice of interface thickness which drives the trade-off between the accuracy of the interface location and the instability of the calculations. In addition, both the CSF and the continuum surface stress (CSS; see last paragraph of this sub-section) formulations of this method tend to produce non-physical velocities near the interface — the spurious currents — due to the numerical imbalance between surface tension and pressure gradient.

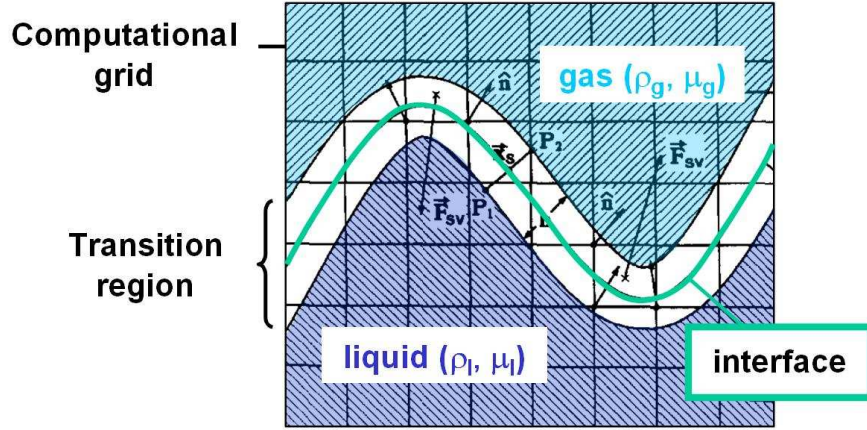


Figure 3.6: Illustration of the continuum surface force method in 2D [26]

Material properties

Assuming the material properties (density, viscosity) to be constant in each phase (see Section 3.1), the equation (3.32) gives the value of a given material property, γ , in the whole computational domain. Using the subscripts *gas* and *liq* to refer to the gas and liquid phases respectively and noting H the Heaviside function, it reads:

$$\gamma(\mathbf{x}) = \gamma_{gas} + H(\mathbf{x} - \mathbf{x}_\Gamma)(\gamma_{liq} - \gamma_{gas}) \quad (3.32)$$

The numerical implementation of (3.32) varies according to the type of scheme used and the method employed to capture the interface. For interface description methods that calculate directly the colour function C (VOF, CLSVOF, CLS, see Section 3.3), the material property considered is then given by:

$$\gamma = \gamma_{gas} + C(\gamma_{liq} - \gamma_{gas}) \quad (3.33)$$

For interface description methods that compute the distance from the interface ϕ , i.e. level set methods, the material properties is given by equation (3.34). Introducing, H_ε , the Heaviside function mollified over a distance proportional to ε , (3.34) reads:

$$\gamma = \gamma_{gas} + H_\varepsilon(\phi - \phi_\Gamma)(\gamma_{liq} - \gamma_{gas}) \quad (3.34)$$

Surface tension forces

In his analysis of blood flows in the heart, Peskin [175] developed the “immersed boundary method” to model numerically the surface force exerted by the heart on the blood. In his framework, the heart wall is modelled by a moving boundary, Γ , “immersed” in a fixed grid domain. This membrane interacts with the fluid through a singular surface force, \mathbf{f}_Γ , approached by a volume force, \mathbf{f}_Ω , spread on a few cells surrounding the “immersed boundary”. Introducing the thickness, h , of the region Ω , surrounding the boundary, then the volume force should give the correct surface force per unit area when $h \rightarrow 0$:

$$\lim_{h \rightarrow 0} \int_{\Delta\Omega} \mathbf{f}_\Omega(\mathbf{x}) \, d\Omega = \int_{\Delta\Gamma} \mathbf{f}_\Gamma(\mathbf{x}_\Gamma) \, d\Gamma \quad (3.35)$$

In the integrals of (3.35), the area $\Delta\Gamma$ represents a portion of the interface and $\Delta\Omega$ is the small volume of thickness h bounding it. In his formulation, Peskin introduces an interpolation function, δ_ε , to mollify the Dirac function on a few cells around the boundary. This convolution kernel satisfies $\delta_\varepsilon \rightarrow \delta_\Gamma$ when $\varepsilon \rightarrow 0$, where ε is a parameter of the kernel function, proportional to h . Using the interpolation function δ_ε , the surface integral of \mathbf{f}_Γ can be approximated by the following volume integral:

$$\int_{\Gamma} \mathbf{f}_\Gamma(\mathbf{x}_\Gamma) \, d\Gamma \approx \int_{\Omega} \mathbf{f}_\Gamma(\mathbf{x}) \delta_\varepsilon(\mathbf{x} - \mathbf{x}_\Gamma) \, d\Omega \quad (3.36)$$

This method has been applied to the numerical handling of surface tension forces by Brackbill et al. [26] in their “continuum surface force” (CSF). Using the CSF model and assuming the surface tension uniform ($\mathbf{f}_\Gamma(\mathbf{x}) = \sigma\kappa(\mathbf{x}) \hat{\mathbf{n}}(\mathbf{x})$), an expression can be found for the capillary forces \mathbf{f}_{cap} in (3.15). When the colour function is directly available, \mathbf{f}_{cap} is approximated by:

$$\mathbf{f}_{cap} \approx \sigma\kappa\nabla C \quad (3.37)$$

Similarly, for level set methods the capillary forces are approximated by:

$$\mathbf{f}_{cap} \approx \sigma\kappa\nabla\phi\delta_\varepsilon(\phi - \phi_\Gamma) \quad (3.38)$$

Continuum surface stress

The CSF method implements the surface tension forces from the formulation of the momentum balance given in (3.15). In (3.15), the surface tension forces are simply added as a source term on the right hand side of the equation after the calculation of the interface curvature. Evolving from the CSF, Lafaurie et al. produced the continuum surface stress (CSS) [119] by expressing the capillary effects as a tensor (3.16). The CSS then models the surface tension by an added correction in the momentum stress tensor, based on the local gradient of the volume fraction.

Due to this tensorial formulation of the capillary effects, the CSS is based on a conservative form of the momentum balance (3.17) and therefore formally conserves the momentum. Although, the CSS results in stronger spurious currents than CSF, unlike CSF, this method tends to reduce the magnitude and the spread of these non-physical velocities under grid refinement [198].

3.4.3 Ghost fluid method

In order to mitigate the limitations of the CSF, Fedkiw et al. [55] proposed a numerical method that retains a sharp interface, respects the jump conditions and significantly reduces the spurious currents. The GFM achieves this by extending continuously the domain of discretisation of each phase into the other phase in a “ghost region”. This region is typically 3 to 5 cells thick, depending on the movement of the interface and on the stencil employed.

Although formally in a single fluid formulation, the GFM involves solving the two phases separately and connecting them through the interfacial jump conditions. In particular, the surface tension is applied via a pressure jump condition (3.39) derived from the normal stress conditions (3.11) in the case of an inert interface (no phase change, no reaction). For a uniform surface tension, the set of jump conditions linking the two phases reduces to:

$$[\rho]_{\Gamma} = \rho_{liq} - \rho_{gas} \quad (3.5)$$

$$[\mu]_{\Gamma} = \mu_{liq} - \mu_{gas} \quad (3.6)$$

$$[p]_{\Gamma} = 2 [\mu]_{\Gamma} \hat{\mathbf{n}} \cdot \mathcal{D} \cdot \hat{\mathbf{n}} - \sigma \kappa \quad (3.39)$$

Once the jump conditions are estimated from the interface characteristics, the method extends continuously the discontinuous quantities in each “ghost region” in order to calculate their discrete derivatives. At each time step, a specific algorithm defines these ghost cells so that the boundary conditions for the Navier-Stokes equations are satisfied. It leads to the simultaneous presence of the two fluids at every grid point in the two “ghost regions”. In this narrow band around the interface, two sets of mass, momentum and energy equations are solved (using standard methods), one for the real fluid and the other for the ghost fluid. The appropriate solution — corresponding to the real fluid at a given location — is then chosen according to the phase marker of the interface description method (e.g. the sign of the distance function for LS method).

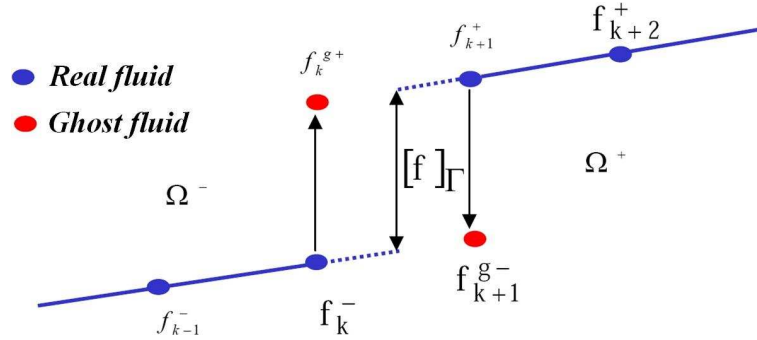


Figure 3.7: Illustration of the ghost fluid method in 1D [246]

The principle of the method in 1D is illustrated in Figure 3.7 for a given variable f discontinuous across the interface Γ . Designating Ω^- and Ω^+ the two domains separated by Γ , the jump of f , $a(x_{\Gamma})$ is defined in (3.40) as a function of $x_{\Gamma} \in \Gamma$. In the LS-GFM framework of Fedkiw et al. [55], $a(x_{\Gamma})$ can be expressed as a function

of the distance from the front ϕ (3.41).

$$[f]_{\Gamma} = f^+ - f^- = a(x_{\Gamma}) \quad (3.40)$$

$$a_{\Gamma} = \frac{a_i |\phi_{i+1}| + a_{i+1} |\phi_i|}{|\phi_i| + |\phi_{i+1}|} \quad (3.41)$$

Using these notations, the ghost values f_{i+1}^{g-} in Ω^- and f_i^{g+} in Ω^+ are defined respectively by (3.42) and (3.43), and the discrete derivative for f is given by (3.44) in cells crossed by the interface [152].

$$f_{i+1}^{g-} = f_{i+1}^+ - a_{\Gamma} \quad (3.42)$$

$$f_i^{g+} = f_i^- + a_{\Gamma} \quad (3.43)$$

$$\left. \frac{\partial f}{\partial x} \right|_{i+\frac{1}{2}} = \frac{f_{i+1} - f_i}{dx} - \frac{a_{\Gamma}}{dx} \quad (3.44)$$

Fedkiw et al. originally combined the GFM with the level set method which calculates directly the distance from the interface ϕ . This interface-capturing method is ideally associated with the GFM as the interface location and the jump conditions, essential to the GFM, can be easily derived from ϕ . As a result, the implementation of the GFM in a level set solver is wieldy: it reduces to the addition of routines to define and treat the ghost cells. In addition, the GFM being applicable to a broad range of jump conditions, the LS-GFM method can be extended to vaporising and reacting two-phase flows [158]. Early implementation of this method for the simulation of incompressible two-phase flows are reported in [109]. Further developments of this technique can be found in [90, 91].

3.4.4 Alternative methods

Immersed interface method (IIM) LeVeque and Li have developed this method to handle jump conditions by applying them directly into the stencil of discretisation [127]. LeVeque and Li combine these jump conditions with Taylor series expansions about the interface to derive their difference scheme. Although this method is precise and reaches second-order accuracy, its algorithms are complex and difficult to extend to 3D. Successful implementation of the IIM for the simulation of incompressible Navier-Stokes equations with immersed boundaries are reported in [126].

“Cut-cell” techniques Popinet and Zaleski developed a 2D front-tracking algorithm (surface markers connected by cubic splines) in order to improve the modelling of surface tension [185]. The authors take advantage of the front-tracking method to produce a novel technique to represent the surface tension based on the idea that the interface is a sharp discontinuity. This technique reduces significantly the spurious currents on the classical test cases (equilibrium bubble, capillary wave and Rayleigh-Taylor instability) and could be extended to the treatment of other discontinuous terms across the interface. However, this method relies on complex geometric considerations and its implementation to 3D is not straightforward.

Parabolic reconstruction of surface tension (PROST) Renardy and Renardy developed a novel technique to handle surface tension by calculating the curvature of the interface from the least square fit of a quadratic approximation of the interface [198]. Thanks to a finer approximation of the curvature, this algorithm dramatically reduces the spurious currents (2 to 3 orders of magnitude lower than with CSF or CSS) and shows better spatial convergence than both CSF and CSS methods. However, such improvement in the accuracy of the method comes at a great computational cost.

3.4.5 Spurious currents

Spurious currents are non-physical velocities — in general vortices — that may occur in the vicinity of the interface when the interfacial forces are modelled. Most numerical methods modelling surface tension produce these undesirable features. These includes front-tracking methods, lattice-Boltzmann methods (in which they were first observed, see [74]) and CSF methods. In their VOF/PLIC implementation of the CSF, Lafaurie et al. [119] measured the order of magnitude of these currents in the case of an equilibrium (static) bubble between two phases of equal density and viscosity (see Figure 3.8). They reported the following relation:

$$u_p \sim 0.01 \frac{\sigma}{\mu} \quad (3.45)$$

In practice, there are essentially two numerical reasons for the occurrence of spurious

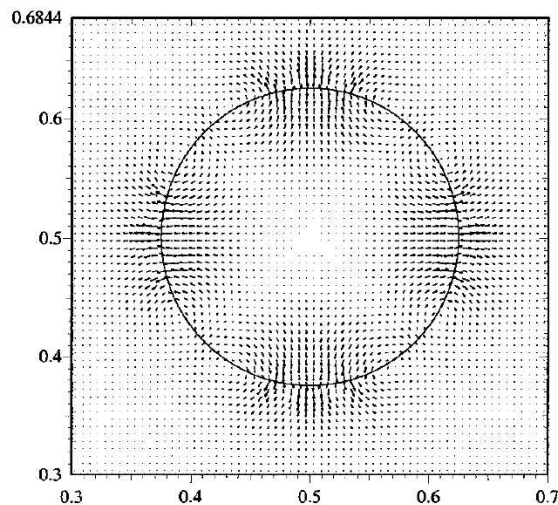


Figure 3.8: Spurious currents in an equilibrium bubble; obtained with the CSF method [198]

currents. The first one is the discrete imbalance between the surface tension and the associated pressure gradient across the interface and the second one is the inaccurate evaluation of the surface curvature.

François et al. mitigated the first problem with the “balance-force algorithm” [59], developed for VOF methods with a CSF model of the interface. This technique eliminates the spurious currents up to machine accuracy (when the curvature is exactly known) by using compatible discrete approximations of the pressure and the volume fraction gradients in the discretised momentum equation. Herrmann applied the same methodology to correct the numerical imbalance in his refined level set grid framework [86]. The LS technique have also been combined with the GFM of Fedkiw et al. [55] and the “sharp-interface method” of Sussman et al. [245] and both achieved very low magnitudes of spurious currents because they apply the jump conditions directly on a sharp front.

As surface tension models are becoming more and more accurate, the reduction of spurious currents relies heavily on the accurate estimation of the curvature. For VOF methods, the difficulty comes from the discontinuity of the volume fraction at the interface location and has been addressed by using convolution operations [273], height-function approaches [83, 183, 240, 245] and high-order interface reconstruction algorithms [183, 198]. With LS methods, the inaccuracies in curvature calculations arise from the fact that the interface does not generally pass through the grid nodes which results in the use of interpolation techniques [137]. Herrmann reached second-order accuracy with his “interface projected curvature evaluation” method [86] based on the distribution of the curvature in the whole computational domain.

In the case of large surface tension and large density ratios, the spurious currents tend to limit the range of capillary number (Ca) and Laplace number (La) that can be simulated accurately, in a similar manner that the Re limits the DNS of homogeneous flows. In particular, above a critical La , corresponding to a large value of σ , the spurious currents dominate the flow and their growth can lead to the destruction of the interface. Introducing the characteristic velocity U , and length

L , the Ca and La are defined in equations (3.46) and (3.47) below:

$$Ca = \frac{\mu U}{\sigma} \quad (3.46)$$

$$La = \frac{\sigma \rho L}{\mu^2} \quad (3.47)$$

Low values of surface tension can also lead to the destruction of the interface. This happens for first-order VOF methods, which are unstable in the absence of surface tension and produce flotsam. Besides, at low Re , a small surface tension leads to a large Ca which results in the creation of cusps in the interface.

This chapter describes the various options available to study the primary breakup with the numerical approach. From this literature review, it appears that the simulation of the primary breakup requires an interface-capturing technique to account for the effect of the geometry on the atomisation process. In particular, the cost-effective conservative level set method, demonstrated on the primary breakup of Diesel jets, appears as a good candidate. In addition, due to the breadth of length and time scales involved in the atomisation process, the LES framework seems necessary to simulate more realistic injection devices.

Chapter 4

Multiphase codes

Two different open source multiphase codes have been identified: Gerris [181] and OpenFOAM [271]. Both codes use VOF to capture the interface and are fairly well-known and recognised in the field of multiphase CFD.

The two codes are briefly described in Section 4.1 and their comparative performance is assessed on typical test cases in Section 4.2. Finally, in Section 4.3, both numerical capabilities are applied to the simulation of the primary breakup.

4.1 Available codes

4.1.1 OpenFOAM

OpenFOAM was originally developed by Henry Weller at Imperial College in London. This code is more a “C++ toolbox” for the numerical solution of continuum mechanics problems than a dedicated CFD solver for multiphase flows. In that regard and thanks to the flexibility given by the C++ language, OpenFOAM is an ideal platform for the customisation and the development of novel CFD methods. OpenFOAM uses finite-volume numerics on unstructured meshes and therefore can handle complex geometries easily. In particular, unstructured meshes can be imported into OpenFOAM from commercial packages (Fluent, Gambit, etc ...).

Thanks to its flexibility and its versatility, OpenFOAM is already a very popular open source code used by several companies and universities around the world. The main issue of the code is its lack of documentation, especially on the implementation of the boundary conditions. Indeed, the official documentation [165, 166] details more the general philosophy of the code than it explains how to use it on a practical problem, as no guidance is given for most of the tutorials. However, some relevant information can be obtained from the “OpenFOAM message board” and from reports or presentations produced by OpenFOAM users.

OpenFOAM already embeds several solvers that handle multiphase flows with a volume of fluid approach. In particular, two different solvers have been identified as relevant test platforms: the DNS solver *interFoam* and the LES version of it *lesInterFoam*.

4.1.2 Gerris

The CNRS (France), and in particular the group lead by Stephane Zaleski, has built up a strong capability in the simulation of multiphase flows. Over the years, this team and its former members (Jie Li, Denis Gueyffier, Stephane Popinet) have developed several codes (SURFER, Gerris) and perfected the VOF method implemented. Stephane Popinet wrote Gerris in C, using quadtree (octree in 3D) finite volume discretisation, to solve mainly multiphase flow problems. This code cannot handle unstructured meshes and treats solid parts as a third phase.

In Gerris, the computational domain is made of 1×1 adjacent boxes, each initially refined to a given level l with: $\Delta x = 1/2^l$. For example a 1×2 domain will be made of two adjacent boxes and a 128×256 mesh will be produced by refining the domain seven times ($2^7 = 128$). The mesh adaptation tool allows the user to set the refinement of each box constituting the domain between two values and according to a set of chosen criteria.

Although fairly unknown in the past few years, Gerris has recently gained in

popularity thanks to the performance of its method. The lack of a user guide is counterbalanced by detailed online tutorials and the profusion of test cases. Further information can be obtained through the message board. The main drawback of this code is the difficulty to develop it, as it was originally written for expert users to be efficient but not “developer friendly”.

This code gathers state-of-the-art numerical methods for VOF interface description: balanced-force CSF formulation, general height-function interface curvature [183] and uses tree-based adaptive mesh refinement [182].

4.2 Code validation

4.2.1 Advection algorithm

The assessment of the advection algorithm of a VOF code requires simulating an interface “transparent” to the flow field. This is achieved by simulating two inviscid phases of identical density with zero surface tension. In the following sub-section, L refers to the length of the computational domain and the error field is calculated by:

$$\Delta C = C_{i,j}^{t=0} - C_{i,j}^{t=1} \quad \forall i, j \quad (4.1)$$

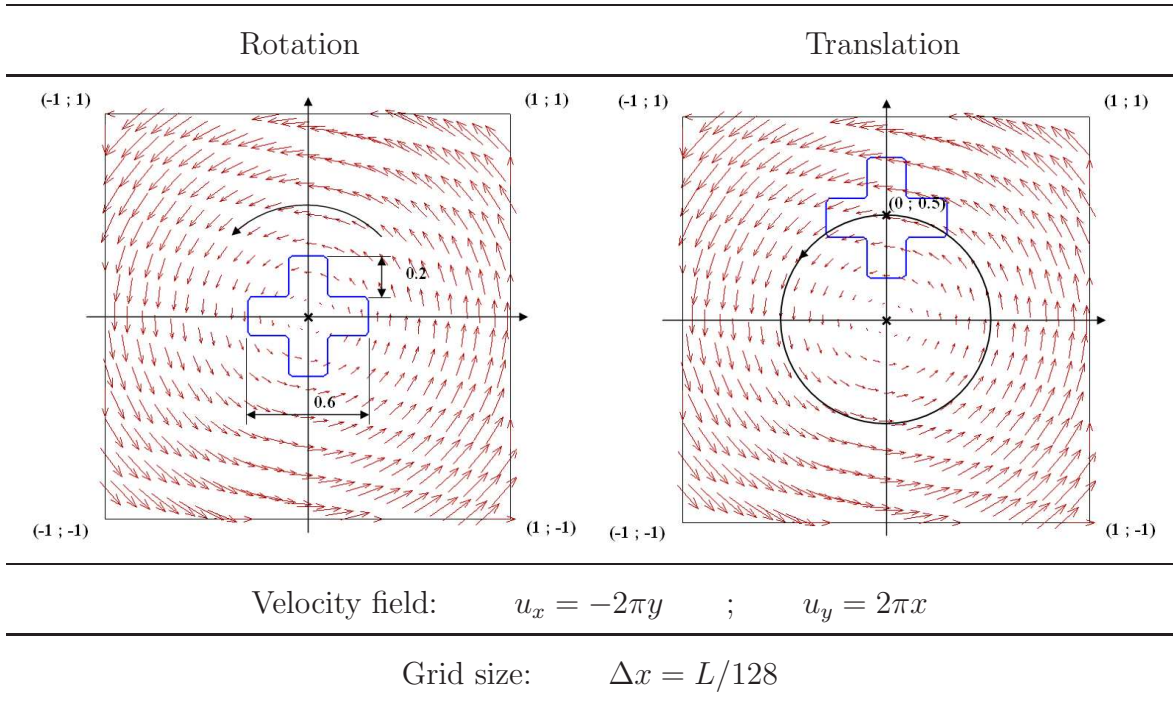
where C is the colour function.

Rotation and translation of a cross The advection of a 2D cross in a solid body rotation field is a simple test case to assess the performance of the interface transport algorithm. The computational domain is a square box delimited by the points $(-1; -1)$ and $(1; 1)$. The cross is made of a 0.6×0.6 square from which four smaller 0.2×0.2 squares are subtracted at each corner (see Table 4.1). The cross has been transported in two different ways with the same velocity field: a pure rotation of the cross about its centre $(0; 0)$, and a translation along a circular trajectory ($C_{traj} = (0; 0)$, $D_{traj} = 1$). The details of both simulations are summarised in

Table 4.1. The end of the simulation is reached after one revolution for $t = 1s$.

The results of the rotation and the translation are respectively reported in Figure 4.1 and Figure 4.2 for both codes. OpenFOAM and Gerris perform the rotation similarly, maintaining the general cross shape. These results compare reasonably well with the review of the existing advection algorithms performed by Pilliod and Puckett [178]. As expected the maximum departures from the original shape occur at the corners. The contours of difference in volume fractions with the original shape are slightly asymmetrical for OpenFOAM. On the translation problem, Gerris' advection algorithm performs equally well and the results present the same pattern as in the rotation case. However, OpenFOAM breaks down on this problem, showing surprisingly bad results not only on the corners of the cross but also along its sides (especially the top of the cross).

Table 4.1: Simulation settings for the advection of a 2D cross



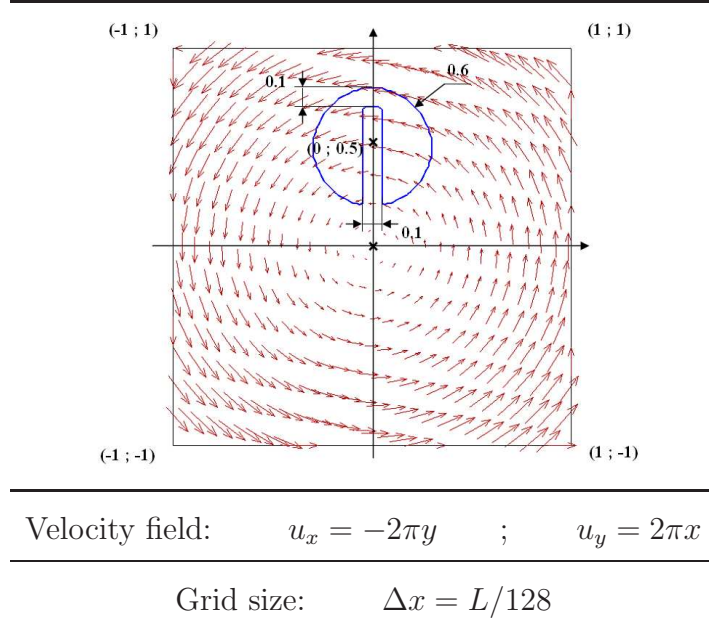
On the translation problem, Gerris has also been run with adaptive mesh refinement based on a curvature criterion (see Table 4.4) ideally suited for such problem. With a level of refinement of 7 ($\Delta x = L/128$). The mesh adaptation tool has been set to refine (resp. coarsen) the mesh up (resp. down) to level 8 (resp. 6) in regions of high (resp. low) curvature. Not only does this feature significantly improve the accuracy of the results but — thanks to the coarsening — it also accelerates the calculation by about 40 %.

The slotted disk of Zalesak Another classic test of the quality of an advection algorithm is the solid body rotation of a slotted disk. This problem, first set by Zalesak [284], is particularly difficult as it involves sharp corners and a thin slot within a solid geometrical shape. Depending on the quality of the scheme, this latter feature may disappear resulting in a modified topology. The computational domain is a square box delimited by the points $(-1; -1)$ and $(1; 1)$. The shape is made of a disk of diameter 0.6, centred on $(0; 0.5)$ from which a vertical rectangle of 0.1×0.5 is subtracted (see Table 4.2). The slotted disk is then translated along a circular trajectory ($C_{traj} = (0; 0)$, $D_{traj} = 1$) like the cross in the previous case. The simulation settings for this problem are given in Table 4.2. The end of the simulation is reached after one revolution for $t = 1s$.

The results, summarised in Figure 4.3, show that both codes maintain the topology of the slotted disk. However, as in the previous test case, Gerris significantly outperforms OpenFOAM. In particular, OpenFOAM produces unexpected wiggles on the top of the disk. Like in the case of the advection of the 2D cross the worst results are obtained on the top of the translated geometry: the region that travels the most.

Disk in a deformation field This test assesses the ability of the code to represent thin ligaments on coarse grids and to avoid the generation of flotsam. In the first half of the simulation (up to $t = 3s$) a 0.3 disk, centred on $(0.5; 0.75)$ in

Table 4.2: Simulation settings for the advection of a slotted disk

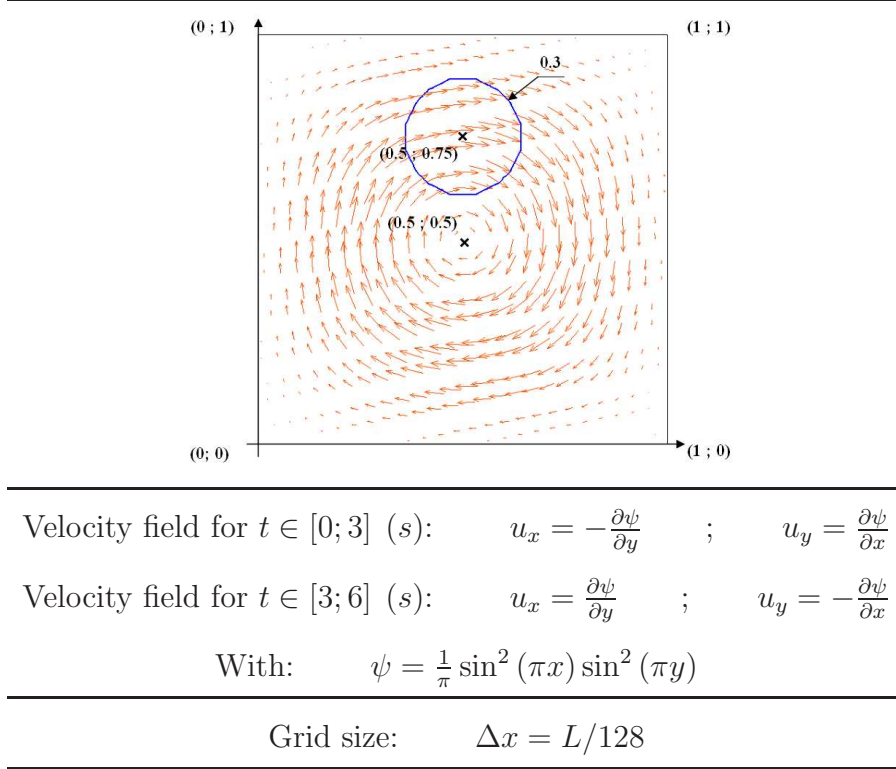


a square box delimited by the points $(0; 0)$ and $(1; 1)$, is deformed into a spiral by a prescribed velocity field. In a second time interval (from $t = 3s$ to $t = 6s$), the opposite velocity field is prescribed so that the original disk is recovered at the end of the calculation (see Table 4.3).

The simulation results, presented in Figure 4.4, highlight the superior performance of Gerris over OpenFOAM. Concerning the representation of thin ligaments, Gerris maintains a continuous spiral for an extra 90° (at $t = 3s$) with respect to OpenFOAM. In addition, at the end of the second phase, Gerris recovers better the initial disk. At the final time, the contours of difference in volume fractions suggest that OpenFOAM does not recover the topology of the disk and creates small liquid structures on the left of the disk.

The adaptive mesh refinement routine of Gerris has been tested with different refinement criteria to highlight the importance of the choice of a criterion. In Table 4.5, the contours of difference in volume fractions are given at the final time,

Table 4.3: Simulation settings for the disk in a deformation field



$t = 6s$, for the simulations without AMR, with adaptation based on the gradient of colour function (C) and with adaptation based on the curvature. Apart from the criterion, the adaptation settings are the same as for the translation of the cross. The results clearly show that for an interface with a lower curvature (except at the tip of the spiral), the adaptation based on the gradient of the color function performs better. Logically, the higher number of cells resulting from this adaptation criterion leads to a longer simulation time.

Table 4.4: Performance of the adaptive mesh refinement on the translation of a 2D cross — Error field, grid size Δx and duration of the calculation t_{calc}

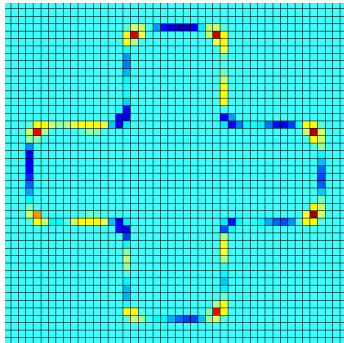
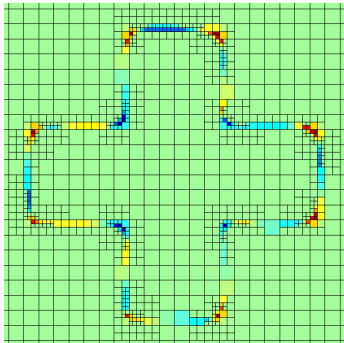
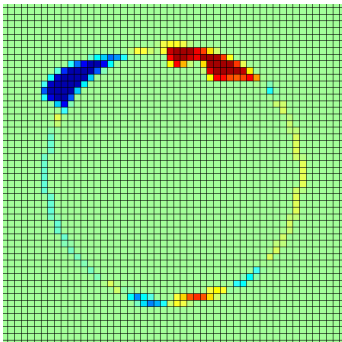
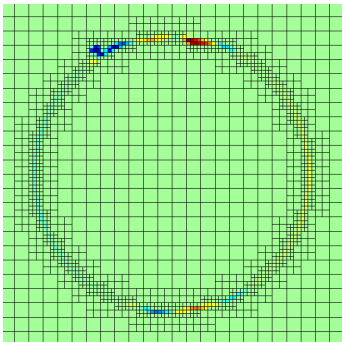
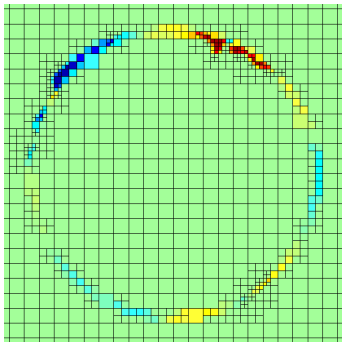
| No adaptation | Adaptation |
|---|--|
|  |  |
| $\Delta x = cst = L/128$ | $\Delta x = L/64 - L/256$ |
| $t_{calc} = 28.1s$ | $t_{calc} = 16.7s$ |

Table 4.5: Performance of the adaptive mesh refinement on the disk in a deformation field — Error field, grid size Δx and duration of the calculation t_{calc}

| No adaptation | Adaptation — ∇C | Adaptation — κ |
|---|---|---|
|  |  |  |
| $\Delta x = cst = L/128$ | $\Delta x = L/64 - L/256$ | $\Delta x = L/64 - L/256$ |
| $t_{calc} = 50.6s$ | $t_{calc} = 77.7s$ | $t_{calc} = 59.2s$ |

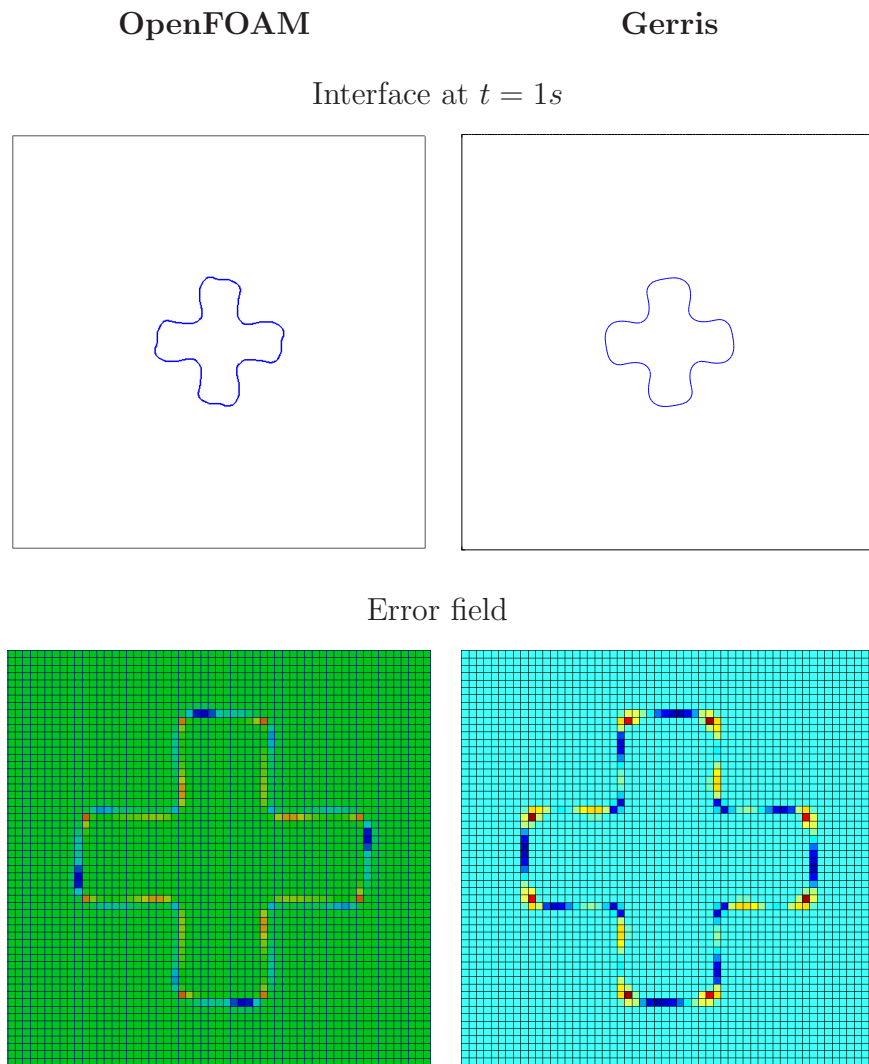


Figure 4.1: Performance of the advection algorithms on the rotation of a 2D cross
— Comparison OpenFOAM vs. Gerris

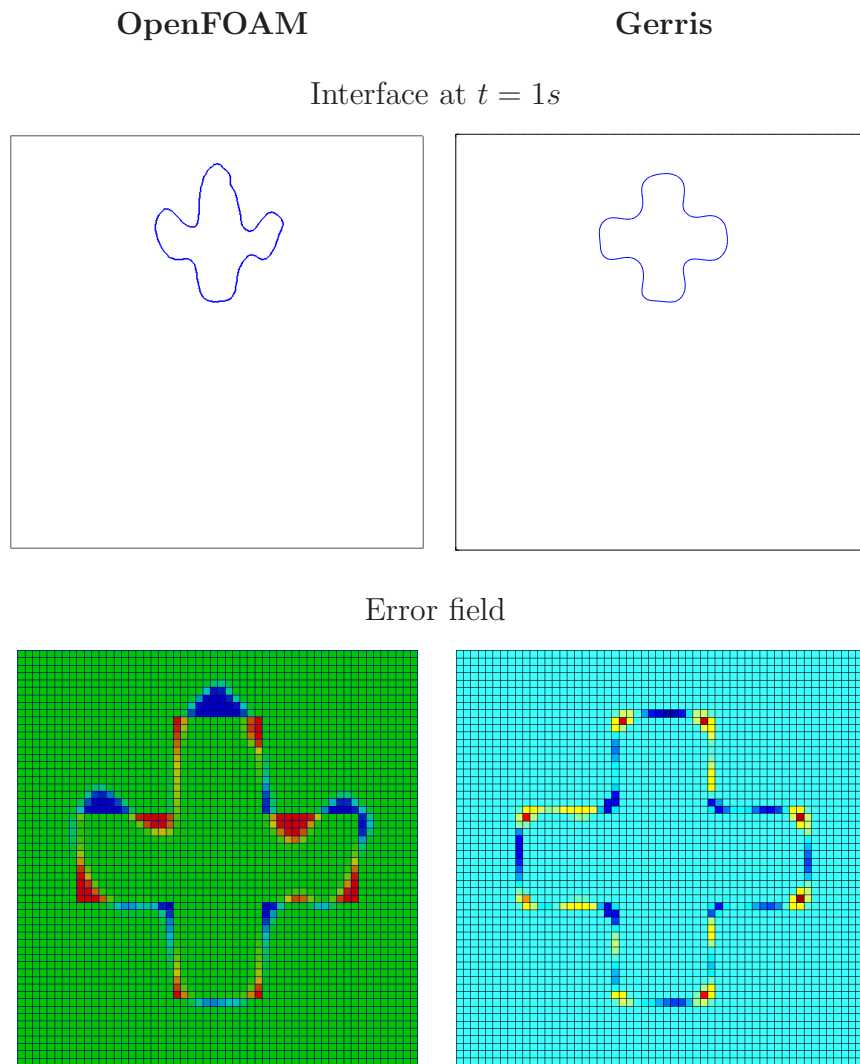


Figure 4.2: Performance of the advection algorithms on the translation of a 2D cross
— Comparison OpenFOAM vs. Gerris

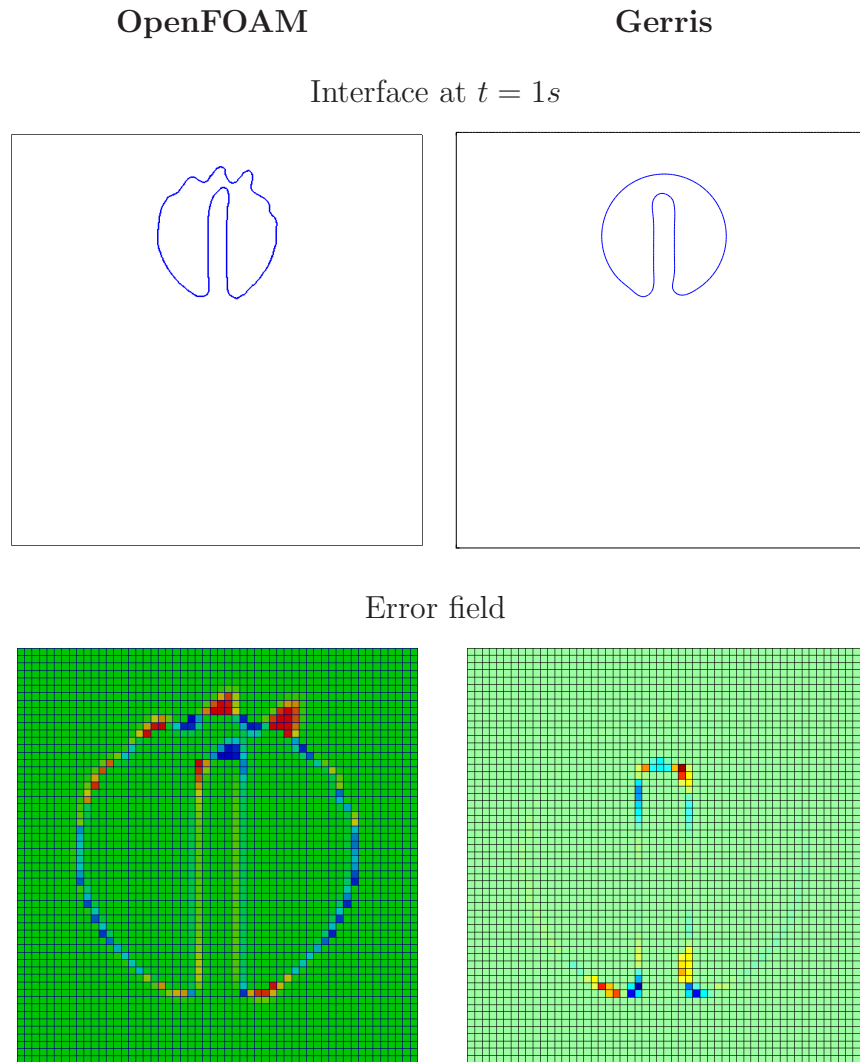


Figure 4.3: Performance of the advection algorithms on the slotted disk of Zalesak
— Comparison OpenFOAM vs. Gerris

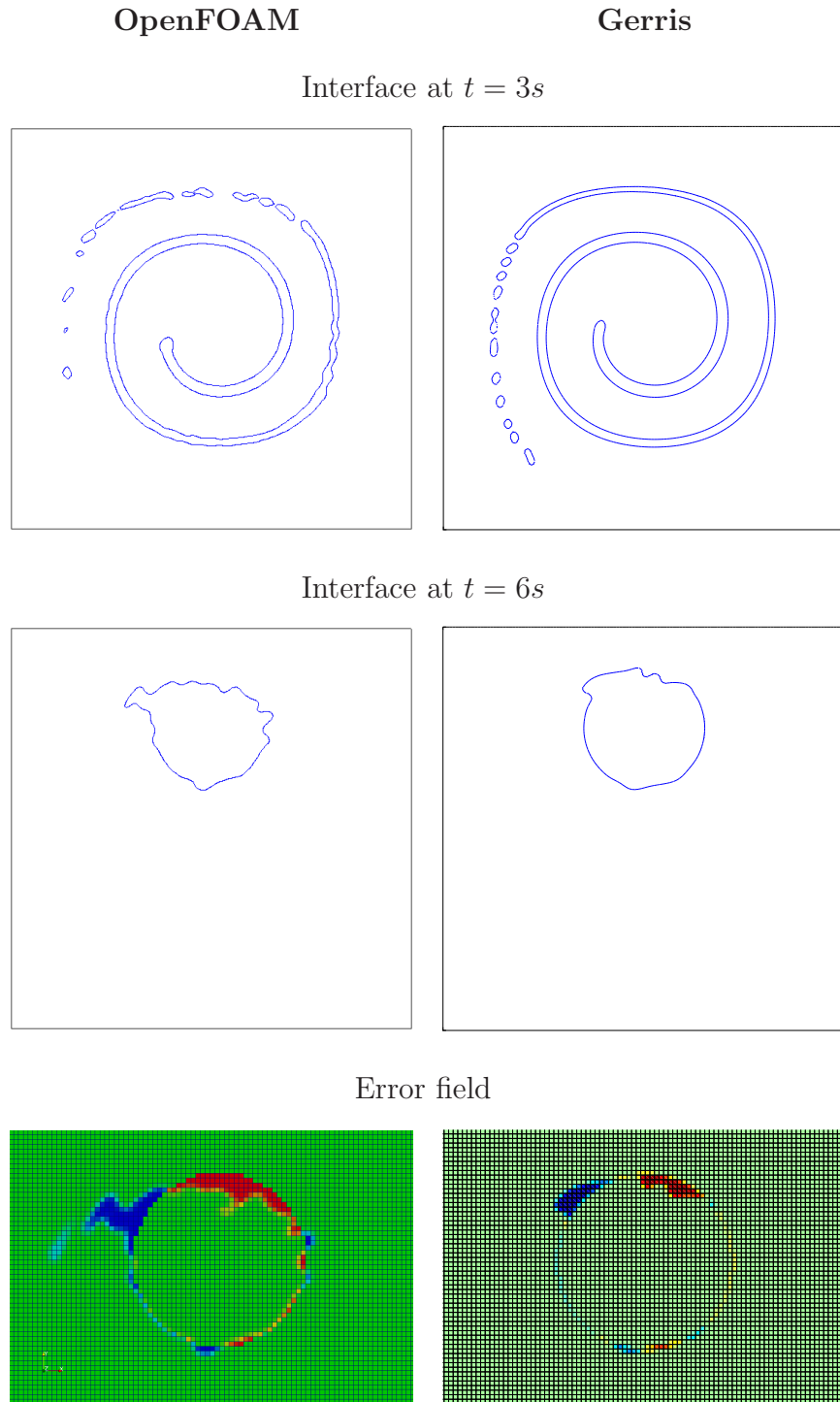


Figure 4.4: Performance of the advection algorithms on a disk in a deformation field
— Comparison OpenFOAM vs. Gerris

4.2.2 Basic two-phase flows

Rayleigh-Taylor instability This test case is usually used to demonstrate the numerical convergence of an interface description method. The Rayleigh-Taylor instability is also used to study the capability of code to capture thin ligaments in the presence of surface tension and with a significant density ratio between phases. Simulations of the Rayleigh-Taylor instability have been carried out with various interface description methods by many authors in both 2D and 3D [133, 185, 189, 260].

The problem is set in a rectangular box (free-slip boundary conditions imposed on all four walls) of $1m \times 4m$ in which two phases of different densities and identical viscosities equally share the domain. The heavier fluid occupies the top half of the domain and is initially separated from the light fluid by a cosine interface of maximum amplitude $5 \times 10^{-2}m$. Under buoyancy effects and because of the original perturbation of the interface, the heavier fluid falls into the light one. The details of the simulation settings are given in Table 4.6.

Table 4.6: Simulation settings for the Rayleigh-Taylor instability

| | |
|-----------------------------|--|
| Initial interface position: | $y = -0.05 \cos(2\pi x)$ |
| Density: | $\rho_1 = 1.255kgm^{-3} \quad \rho_2 = 0.1694kgm^{-3}$ |
| Viscosity: | $\mu_1 = \mu_2 = 3.13 \times 10^{-3}kgm^{-1}s^{-1}$ |
| Surface tension: | $\sigma = 1 \times 10^{-2}Nm^{-1}$ |
| Gravity: | $g = 9.81ms^{-2}$ |
| Mesh size: | 128×512 |

The problem, as formulated in [189] has been simulated with OpenFOAM and Gerris. The results of both codes, presented in Figure 4.5, compares satisfactorily with the calculations presented in [133, 185, 189]. However, as demonstrated in the

previous section, Gerris captures better the thin liquid structures. In particular, a close up in the region of the “mushroom cap” — formed by the penetration of the denser phase — highlights significant differences on the outer edge of the mushroom cap at $t = 1s$.

The simulation on a finer mesh (256×1024) with OpenFOAM confirms the higher accuracy of Gerris (see Figure 4.6). Also, the comparison of the two OpenFOAM simulations (128×512 vs. 256×1024) suggests that the correct capture of the thin ligaments has a strong influence on the overall simulation results: lower penetration of the dense phase and wider opening of the mushroom cap. In addition, OpenFOAM seems to create wiggles in the neck of the stalk close to the cap. These waves on the interface are not perceptible in other calculations (Gerris and [133, 185, 189]) and seem to shrink — in both amplitude and wavelength — with mesh refinement.

This problem has been run with “adaptive mesh coarsening” based on two criteria: the vorticity and the gradient of the colour function. In Figure 4.7, the meshes obtained with and without AMR are represented in parallel to the contours of colour function and vorticity, so that the link between the pattern formed by the adapted mesh and the adaptation criterion appears clearly. Because the AMR is based on an octree architecture, the same accuracy of interface description is guaranteed but at much lower computational cost. In particular, this “adaptive mesh coarsening” divides the number of cells by 2.7 and the simulation time by 7.

Falling drop in a pool This test case was set by Sussman et al. in [241] in order to test their level set - AMR framework. It consists in simulating the “splash” produced by an axisymmetric droplet impacting a liquid surface in the case of a large density ratio (water and air). This simulation involves both merging and pinching of interfacial surfaces with respectively the impact of the droplet on the pool interface and the subsequent splash.

The domain is a $2mm \times 4mm$ box set with the following boundary conditions: axisymmetry on the left wall, free-slip on the bottom and the right wall and outflow

on the top wall. The $1mm$ water drop is projected — from an initial height of $0.5mm$ — with a velocity of $4ms^{-1}$ towards a pool of water that fills the bottom half of the domain. The details of the simulation settings are presented in Table 4.7.

Table 4.7: Simulation settings for the falling drop in a pool

| | | |
|---------------------|--|---|
| Droplet parameters: | $\phi_{drop} = 1 \times 10^{-3}m$ | $\mathbf{u} = (0 \ -4 \ 0)(ms^{-1})$ |
| Density: | $\rho_1 = 1000kgm^{-3}$ | $\rho_2 = 1.225kgm^{-3}$ |
| Viscosity: | $\mu_1 = 1.137 \times 10^{-3}kgm^{-1}s^{-1}$ | $\mu_2 = 1.78 \times 10^{-5}kgm^{-1}s^{-1}$ |
| Surface tension: | $\sigma = 7.28 \times 10^{-2}Nm^{-1}$ | |
| Reynolds number: | $Re = 3520$ | |
| Weber number: | $We = 220$ | |
| Mesh size: | 128×256 | |

The contours of volume fractions predicted by Gerris and OpenFOAM are presented in Figure 4.8 for five time steps ($t \in [0; 0.55] ms$). The simulation results compare well with the contours in [241] and the two codes provide the same interface topology for most of the time steps. However, once again, Gerris seems to treat better the small inclusions of one phase into the other: thin liquid structures and bubbles. In particular, at $t = 0.2ms$ whereas OpenFOAM has already separated the coronet from the bulk of the liquid, Gerris has maintained the integrity of the liquid structure. Also, from $t = 0.15ms$ to $t = 0.55ms$, although both codes represent the toroidal bubbles produced at the impact of the drop on the surface, OpenFOAM seems to diffuse them rapidly whereas Gerris maintains their integrity on the same mesh. It is hypothesized that OpenFOAM’s diffusion of the small structures (e.g. resulting in the loss of thin ligaments) is related to the advection algorithm. Indeed, the poor performance of this scheme, highlighted in Section 4.2, suggests that OpenFOAM struggles to maintain the topology and the integrity of structures when

transported over relatively long distances.

Phase inversion This problem was set by Labourasse et al. in [118] in order to test the capability of a method to simulate strongly deformed dynamic interfaces in view of predicting turbulent two-phase flows. In addition, because of the existence of a theoretical solution, this problem also assesses the mass conservation properties of the technique tested.

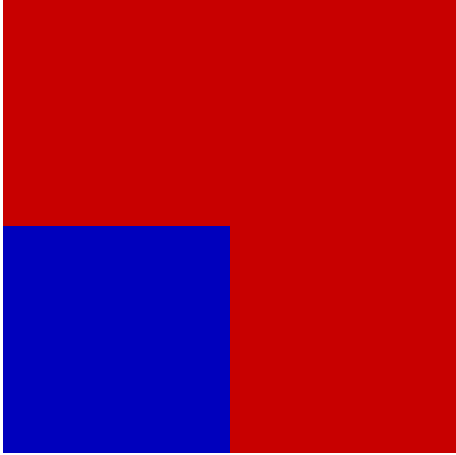
The problem is set in a square box (free-slip boundary conditions imposed on all four walls) of $1m \times 1m$. The bottom left-hand corner of the box is filled with a square $0.5m \times 0.5m$ inclusion of oil while the rest of the domain ($3/4$ of the box) is filled with water (see Table 4.8). Under buoyancy effects, the oil phase reaches the top quarter of the domain. The material and environmental properties are such that the surface tension forces are dominating and the oil progresses to its final position — the top quarter of the computational domain — through a series of steps involving strong deformations of the interface and numerous topology changes.

The contours of volume fractions predicted by Gerris (mesh: 128^2) and by OpenFOAM (meshes: 128^2 and 256^2) are presented in Figure 4.9, Figure 4.10 and Figure 4.11 for various time steps ($t \in [1; 100]$ s). The simulation results compare reasonably well with the contours in [118]. Some differences between Gerris' and OpenFOAM's predictions are perceptible on the simulation results obtained with the 128^2 mesh. These differences get stronger as the interface is more and more distorted ($t \in [6.25; 22.25]$ s), and then fade out as the interface topology gets simple again ($t \in [34.75; 100]$ s).

This problem presents some “chaotic behaviour” in terms of interface topology, since a small difference at the beginning of the simulation leads to a completely distinct pattern when the distribution of phases is the most homogeneous. The buoyancy effects finally assist the convergence of the phase distribution, for all simulations, towards the simple pattern of the theoretical solution.

In the early stages of the simulation ($t \in [1; 6.25]$ s) the pattern of the multi-

Table 4.8: Simulation settings for the phase inversion

| Oil inclusion in water | | $(L_{oil} = 0.5m)$ |
|--|--|--|
|  | | |
| Density: | $\rho_1 = 1000kgm^{-3}$ | $\rho_2 = 900kgm^{-3}$ |
| Viscosity: | $\mu_1 = 1 \times 10^{-3}kgm^{-1}s^{-1}$ | $\mu_2 = 1 \times 10^{-1}kgm^{-1}s^{-1}$ |
| Surface tension: | $\sigma = 4.5 \times 10^{-2}Nm^{-1}$ | |
| Gravity: | $g = 9.81ms^{-2}$ | |

phase flow predicted by OpenFOAM on the refined mesh seems to be halfway between OpenFOAM's and Gerris' on the coarse mesh. This suggests that Gerris also performs better on the phase inversion problem.

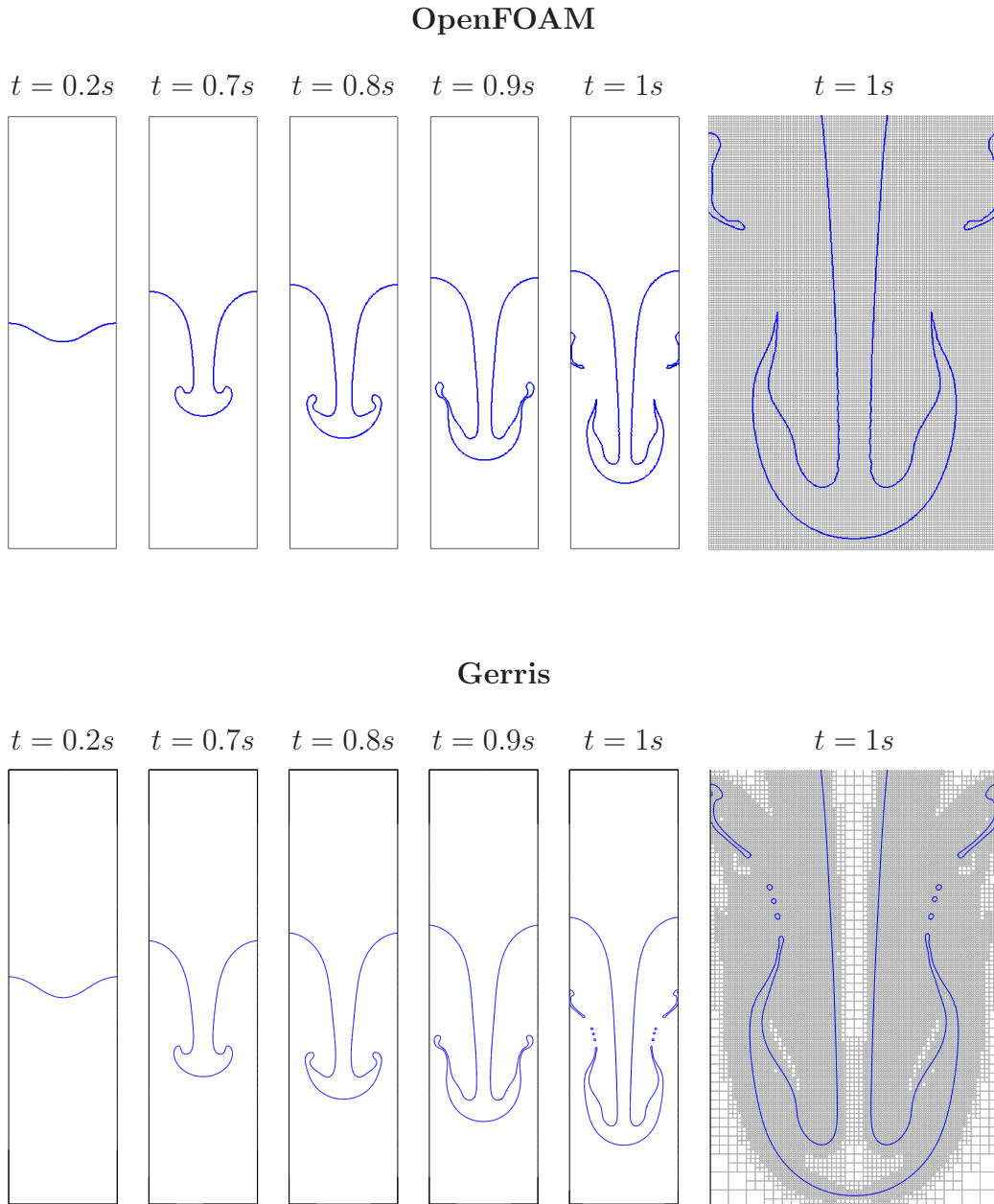


Figure 4.5: Simulation results for the Rayleigh-Taylor instability — Interface predicted by OpenFOAM (top) and Gerris (bottom)

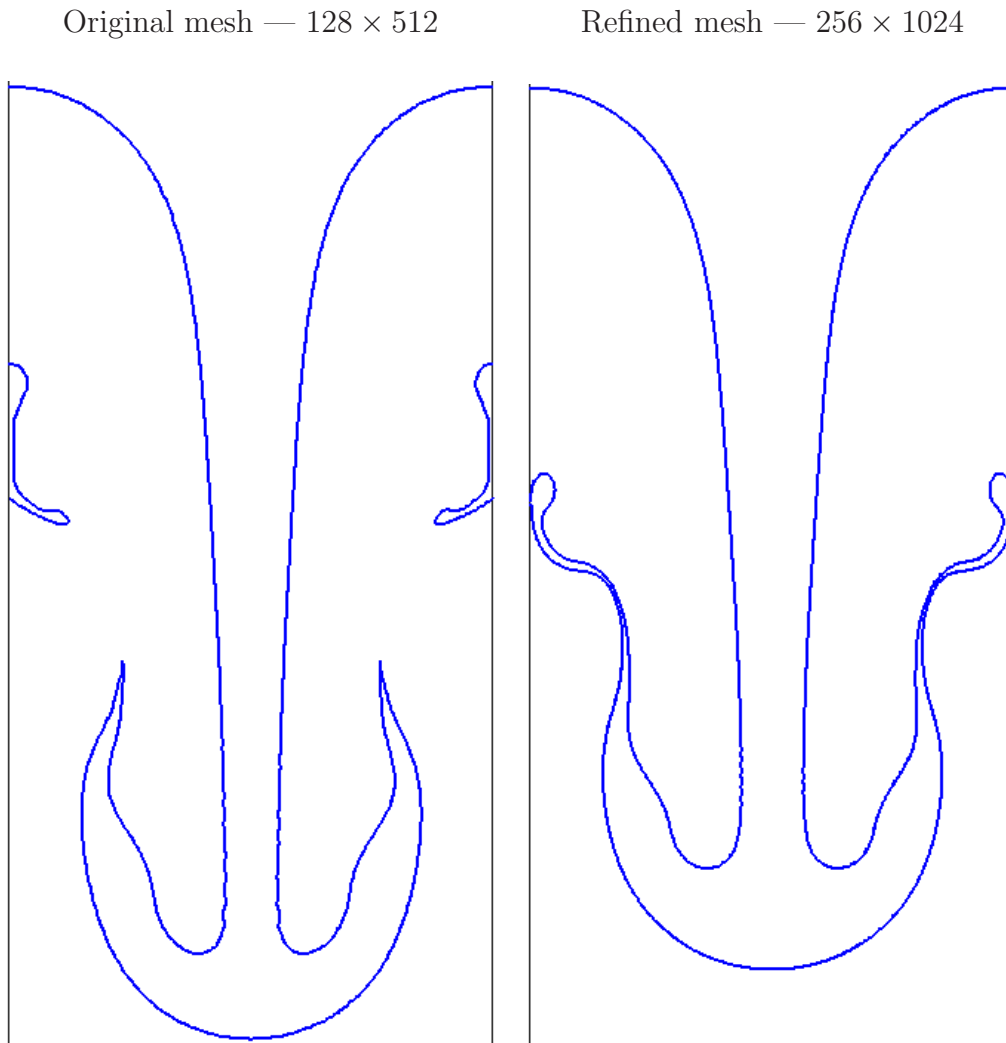


Figure 4.6: Performance of OpenFOAM on the simulation of the Rayleigh-Taylor instability — Detail of the interface in the region of the neck at $t = 1s$ for meshes: 128×512 ($t_{calc} = 6126s$) and 256×1024 ($t_{calc} = 35688s$)

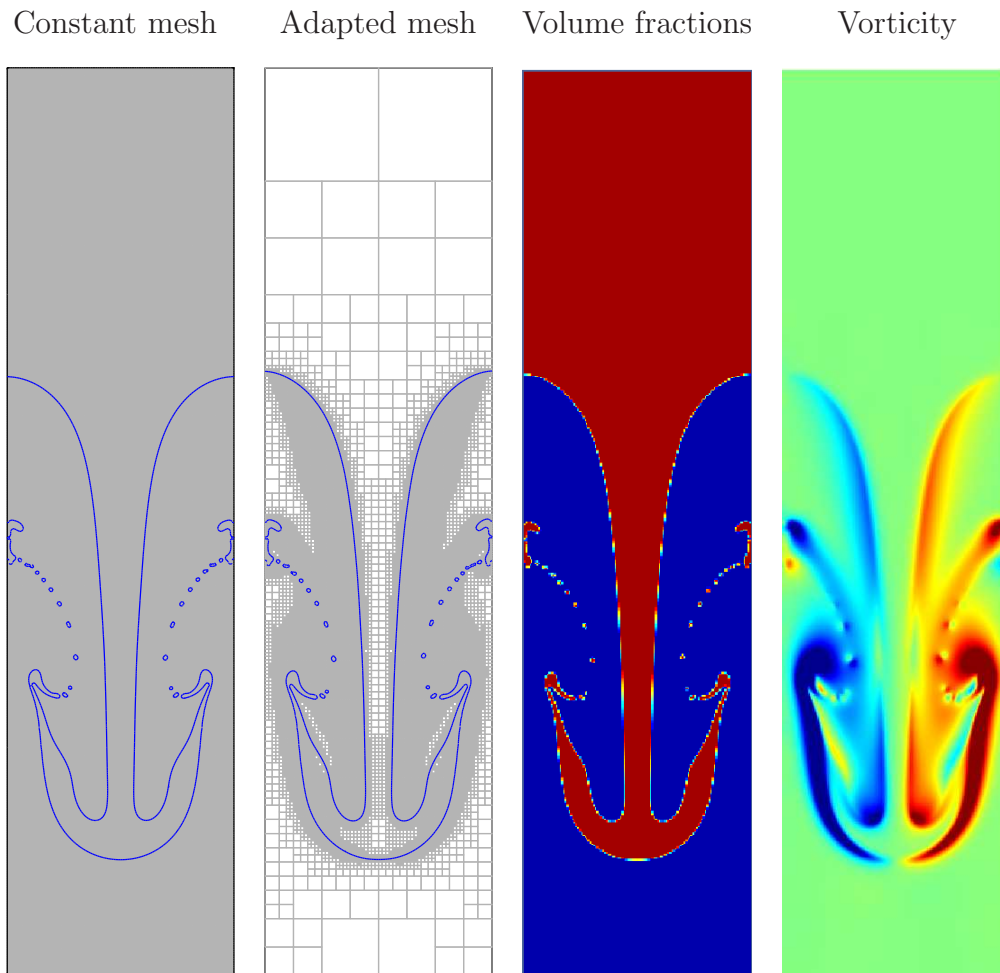


Figure 4.7: Performance of Gerris on the simulation of the Rayleigh-Taylor instability without surface tension — Results at $t = 1s$ for constant mesh ($N_{cells} = 65536$; $t_{calc} = 1338s$) and adapted mesh with $l_{max} = 7$ ($N_{cells} = 24274$; $t_{calc} = 193s$)

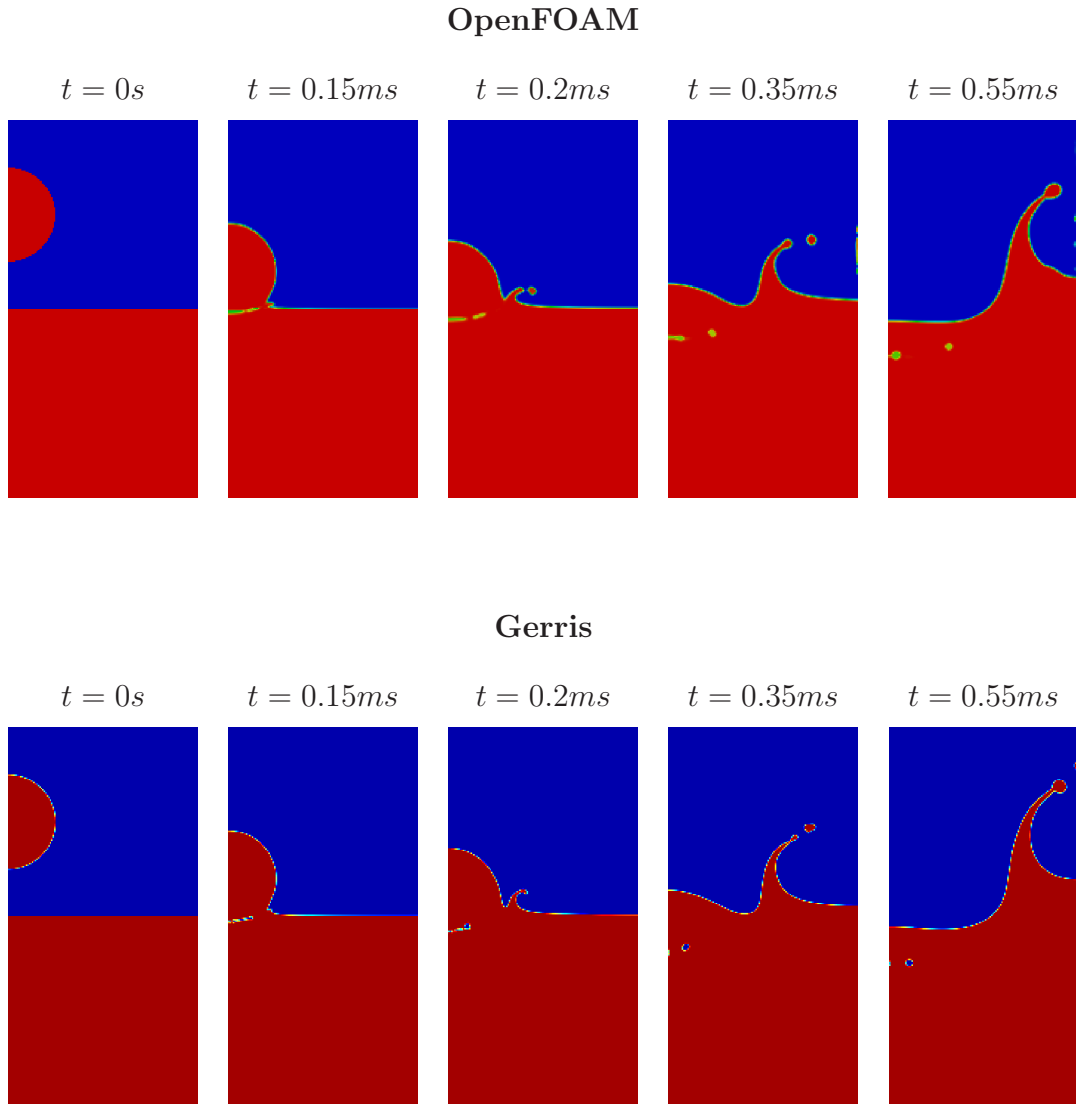


Figure 4.8: Volume fractions predicted by the codes for the falling drop in a pool
 — Volume fractions predicted OpenFOAM (top) and Gerris (bottom)

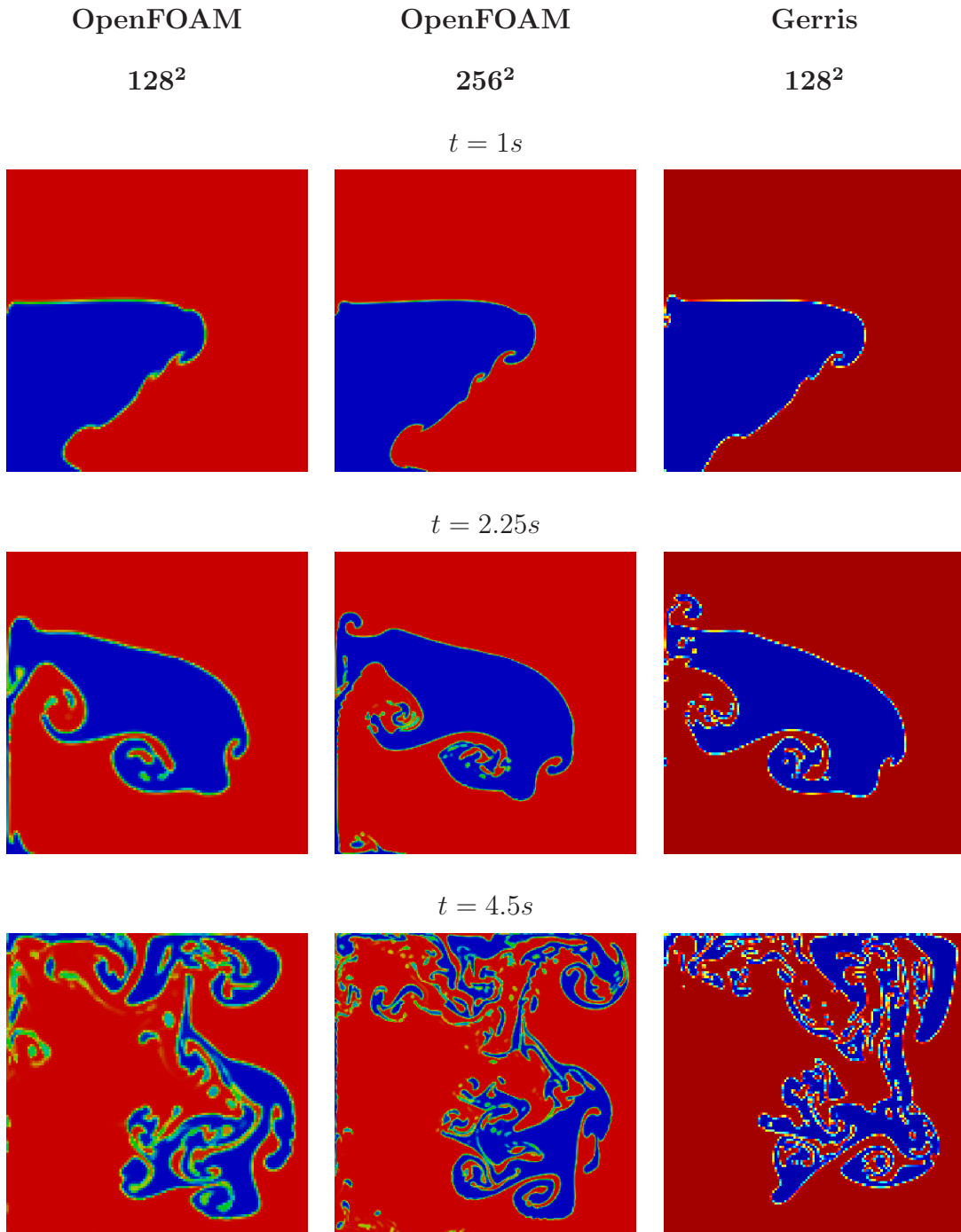


Figure 4.9: Volume fractions predicted by the codes for the phase inversion — Time $t = 1s, 2.25s, 4.5s$

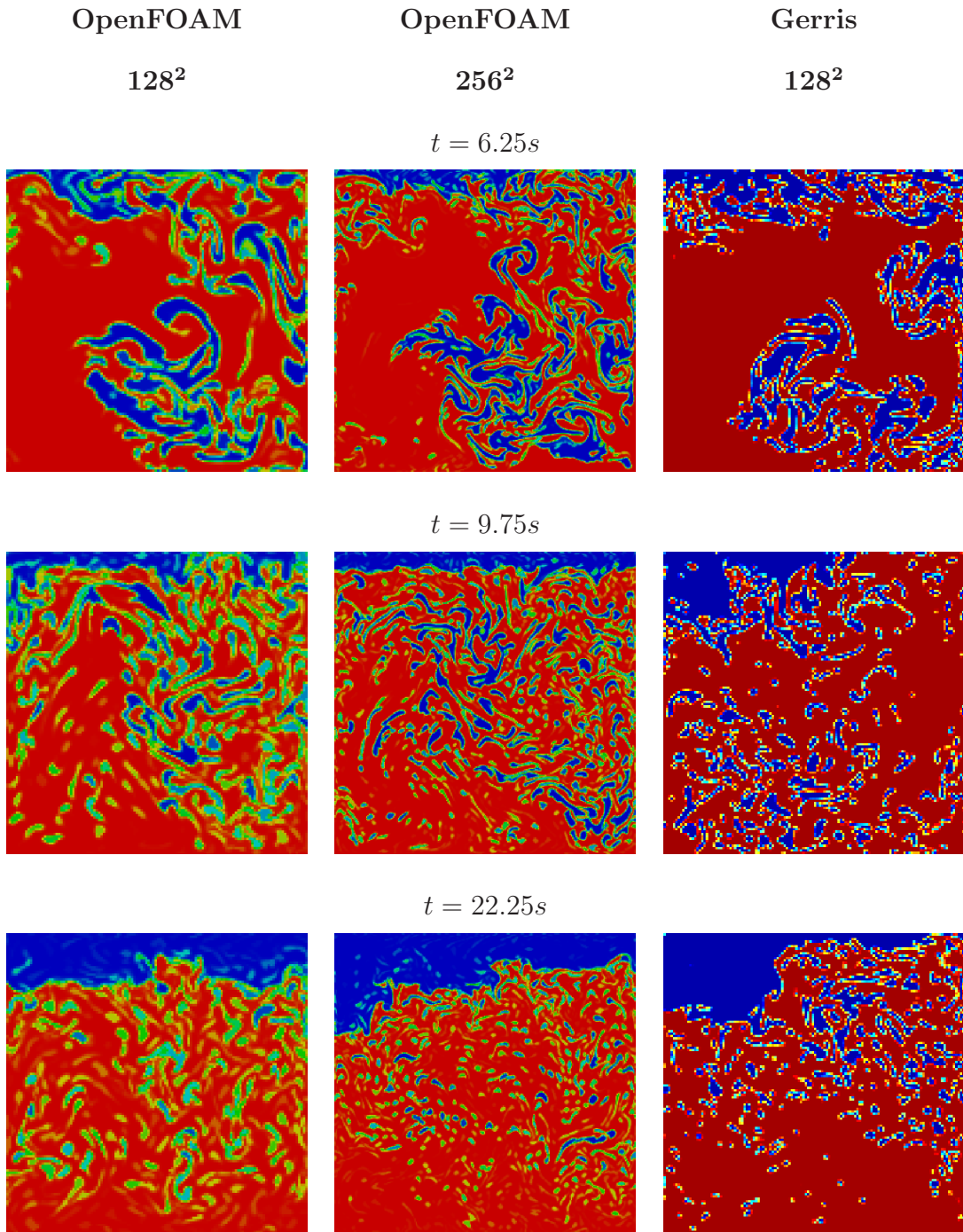


Figure 4.10: Volume fractions predicted by the codes for the phase inversion —
Time $t = 6.25s, 9.75s, 22.25s$

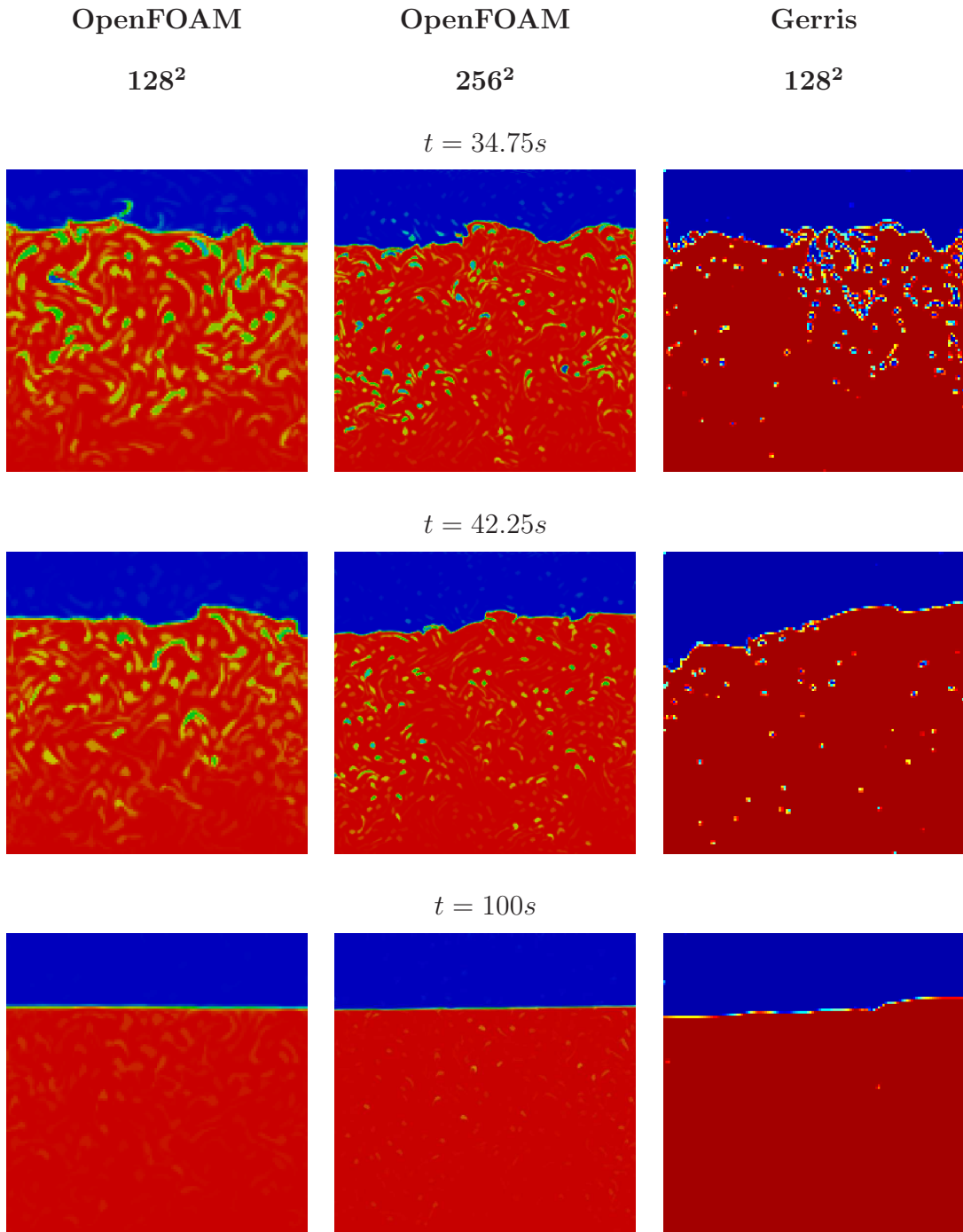


Figure 4.11: Volume fractions predicted by the codes for the phase inversion —
Time $t = 34.75s, 42.25s, 100s$

4.3 Simulation of atomisation

Two types of simulation have been considered in order to take advantage of the best features of both codes: the simulation of atomisation with sub-grid scale modelling, using OpenFOAM (*lesInterFoam*, see [44] for details on SGS), and without, using Gerris. The settings of all spray calculations are presented in tables 4.9, 4.10 and 4.11, and the corresponding non-dimensional numbers are given in Table 4.12.

The simulation of liquid breakup with SGS models involved reproducing the results of Villiers et al. on the atomisation of Diesel jet [44] and extending this approach (similar physical properties and domain dimensions) to the breakup of an axisymmetric sheet. The latter calculation was done on a very coarse mesh as a mean to illustrate the capability of OpenFOAM.

The simulations of flat sheet breakup without SGS models have been run using Gerris with similar domain dimensions, fluid densities and surface tensions. However, in order to illustrate the effect of We on the spray, the viscosity has been reduced for both fluids. Indeed, the turbulence in the flow field has a strong effect on the primary breakup [54, 146] and to obtain a spray in a smaller computational domain, we have artificially increase the Re in both phases. Also, as demonstrated in experimental studies [135, 136, 145, 146, 195], thinning the vorticity layer increases the growth rate of the interfacial instabilities and therefore reduces the breakup length. The effect of the Weber number on the breakup (see Section 2.4) has been illustrated on three simulations of increasing We . For the highest We , the build-up of instabilities and the onset of the breakup are presented.

Due to the significant influence of turbulence build-up in the injection pipe [54, 146], the flow in the injection channel has been modelled for a distance of three to five injection hole characteristic lengths depending on the configuration of the spray calculation.

Table 4.9: Physical properties for the spray calculations with SGS models

| | | Fuel | Gas | Ratio fuel/gas |
|---------------------------|--------------------|-----------------------|----------------------|----------------|
| Density, ρ | (kgm^{-3}) | 840 | 20 | 42 |
| Viscosity, μ | $(kgm^{-1}s^{-1})$ | 5×10^{-3} | 1.7×10^{-5} | 294 |
| Surface tension, σ | (Nm^{-1}) | 2.61×10^{-2} | | |

Table 4.10: Physical properties for the spray calculations without SGS models

| | | Fuel | Gas | Ratio fuel/gas |
|---------------------------|--------------------|------------------------|----------------------|----------------|
| Density, ρ | (kgm^{-3}) | 840 | 20 | 42 |
| Viscosity, μ | $(kgm^{-1}s^{-1})$ | 5.952×10^{-6} | 8.5×10^{-7} | 7 |
| Surface tension, σ | (Nm^{-1}) | 2.61×10^{-2} | | |

Table 4.11: Geometrical parameters for all the spray calculations in Section 4.3

| | Diesel jet | Curved sheet | Flat sheet |
|-----------------------------|---------------------|---------------------|---------------------|
| Configuration | cylindrical | annular | prismatic |
| Nozzle outlet shape | disk (d) | annulus (t) | rectangle (h) |
| Nozzle outlet dimension | $2 \times 10^{-4}m$ | $2 \times 10^{-4}m$ | $2 \times 10^{-4}m$ |
| Separating plates thickness | — | $2 \times 10^{-5}m$ | $2 \times 10^{-5}m$ |
| Domain tangential extent | 90° sector | 90° sector | width $W = 5h$ |
| Domain radial extent | diameter $D = 5d$ | thickness $T = 5t$ | height $H = 5h$ |
| Channel length | $3d$ | $3t$ | $5h$ |
| Domain length | $13d$ | $13t$ | $15h$ |

Table 4.12: Non-dimensional numbers for all the spray calculations in Section 4.3

| | Diesel jet | Curved sheet | Flat sheet | | |
|------------|-----------------------|-----------------------|----------------------|----------------------|----------------------|
| Re_{liq} | 15500 | 67 | 56500 | 56500 | 56500 |
| Re_{gas} | — | 18800 | 94100 | 188000 | 377000 |
| We | 1.36×10^6 | 9300 | 410 | 2100 | 9300 |
| Oh | 7.55×10^{-2} | 7.55×10^{-2} | 9.0×10^{-5} | 9.0×10^{-5} | 9.0×10^{-5} |

4.3.1 Simulations of liquid breakup with SGS model

The two geometries modelled with SGS models involve an axisymmetric configuration (jet and annular sheet breakup). In order to keep the computational cost reasonable, only a 90° sector has been modelled. No turbulence boundary conditions have been set at the injection of the phases: it has been assumed that the appropriate level of turbulence would be developed in the channels modelled. This assumption would require sensitivity studies on the length of the pipe modelled and the results should be compared with the outcome of the method of Klein et al. [112] to generate velocity data artificially.

Diesel jet breakup Like all the simulation settings of this case, the mesh in the radial direction is identical to the one shown in [44]. It can be found in one of the tutorials of the *lesInterFoam* solver for a 2D geometry. The total number of cells of this 90° sector (see Figure 4.12) is halfway between the coarse and fine meshes in [44]. The overall refinement strategy for this mesh is geared to accommodate the formation of small liquid structures at the interface. In particular, the highest level of refinement is at the radius of the injection hole, in the vicinity of the nozzle outlet to represent at best the instabilities triggering the onset of the breakup.

The simulation results have been commented and analysed in detail in [44] and their validity is considered in Section 5.1.1. Only the first stages of the injection, corresponding to the onset of the breakup, are presented in Figure 4.13 below.

The build-up of longitudinal waves along the interface is apparent in the injection channel ($t = 1 \times 10^{-6}s$) and, once the jet has passed the nozzle outlet, in the vicinity of the nozzle. However, the expected transverse instabilities, due to a Rayleigh-Taylor type of instability [146], do not appear clearly until $t = 3.508 \times 10^{-6}s$ when thin ligaments start to form. This delay in the growth of these instabilities could be due to incorrect levels of turbulence in the liquid or a lack of resolution in the azimuthal direction.

In addition, at $t = 1 \times 10^{-6}s$, some “dimples” are present along the interface at the tip of the liquid jet. However, these features seem to follow the mesh too closely and are certainly a numerical artefact due to the lack of resolution in the channel. Further simulation on a finer mesh should address this question.

As explained in [44], the modelling of a sector when simulating jet breakup prevents the jet to flap and consequently extends significantly the jet core length. In particular, the large droplet at the tip of the jet ($t = 3 \times 10^{-6}s$) would have broken off earlier in a full 3D simulation.

The expected gradual rise in vorticity magnitude and its spread is apparent on the simulation results. This highlights the strong interaction between the turbulence and the liquid structures and emphasised the need to develop sub-grid scale models to better capture this phenomenon.

Axisymmetric sheet breakup This case is simply the extension of the Diesel jet problem to an axisymmetric sheet breakup. The extent of the computational domain is based on the same approach and the mesh has been designed with the same philosophy (see Figure 4.14). In order to account for the effect of the separator plates on the onset of the breakup (see [65]), the geometry modelled comprises separator plates of finite thickness around which the mesh is particularly refined. This case has been run in order to demonstrate OpenFOAM’s capability to simulate sheet breakup. The mesh produced being very coarse, the analysis of the calculation should be understood as provisional until further mesh convergence studies confirm these results. No quantitative results should be expected from this simulation.

The same physical properties have been taken for the phases in presence but their condition of injection have been modified to come closer to aero-engine conditions. In particular, the gas stream is co-flowing ($\|\mathbf{u}_{gas}\| = 40ms^{-1}$) on either side of the liquid sheet such that the atomisation is assisted. Also, the velocity of the fuel injection has been reduced from $\|\mathbf{u}_{liq}\| = 460ms^{-1}$ to $\|\mathbf{u}_{liq}\| = 2ms^{-1}$. The resulting non-dimensional number (We and Re_{liq}) are significantly lower than for the Diesel

jet case. The use of the exact same properties between the two cases allows direct comparison of the performance of the injection systems. Besides, the properties of the two phases in the Diesel jet injection and the aero-engine application, although different, are of the same order of magnitude.

The earliest stages of the sheet atomisation are presented in Figure 4.15, from the growth of interfacial waves, in the injection channel, to the onset of the breakup. Longitudinal instabilities are clearly apparent on the simulation results from $t = 4.5 \times 10^{-4} s$, on top of which transverse instabilities appear around $t = 5.5 \times 10^{-4} s$.

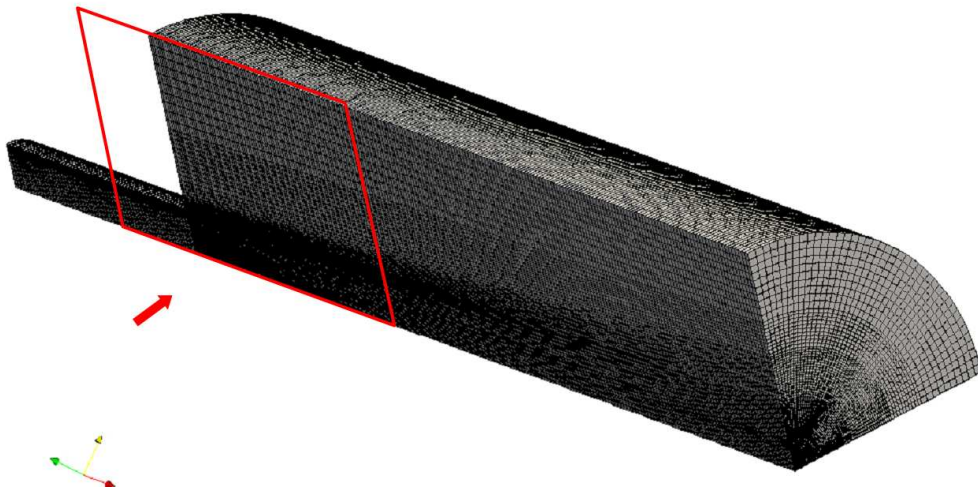
Besides, the pattern of the vorticity contours originally ($t = 1.54 \times 10^{-4} s$) resembles Von Karman streets as the gas is injected on either sides of the liquid injection “crown”. The vorticity contours then evolve to a less structured pattern ($t = 3 \times 10^{-4} s$) and the turbulent structures seem to interact with (potentially initiate) the flapping of the liquid sheet.

Two different breakup mechanisms have been identified in this calculation: one resulting from the effect of the longitudinal wave (mechanism A, see Figure 4.16) and the other one due the combined effect of both longitudinal and azimuthal instabilities (mechanism B, see Figure 4.17).

The first mechanism involves the breakup of large liquid structures at the peak of a longitudinal wave which then further fragments into streamwise ligaments, themselves breaking up into droplets under Plateau-Rayleigh instability.

The second mechanism leads to the formation of droplets through the combined effect of instabilities in both longitudinal and transverse directions. The orthogonal undulations stretch, and eventually tear, the liquid sheet at their peaks, leaving liquid blobs at their nodes. This process is illustrated Figure 4.17, where the transverse instability first tears the flapping liquid sheet in the longitudinal direction. Then, a longitudinal wave, superimposed on the flapping mode, breaks up the large longitudinal structures into droplets.

Isometric view of the domain — 6.83×10^5 cells



Detail of the nozzle outlet region

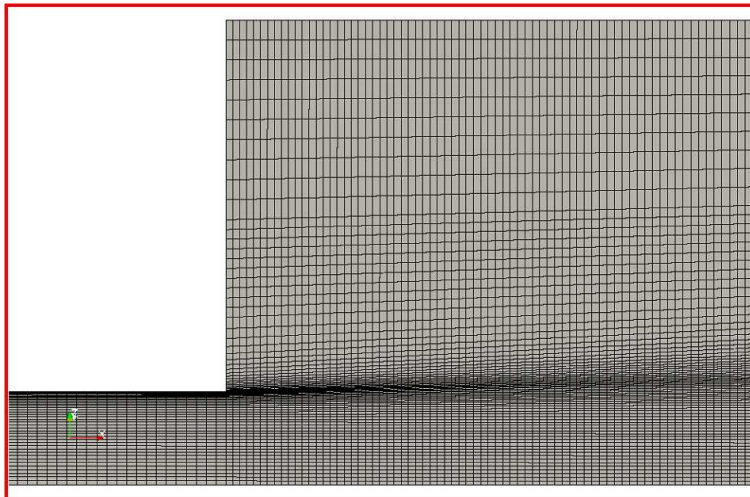


Figure 4.12: Simulation of Diesel jet breakup with OpenFOAM — Mesh of the domain

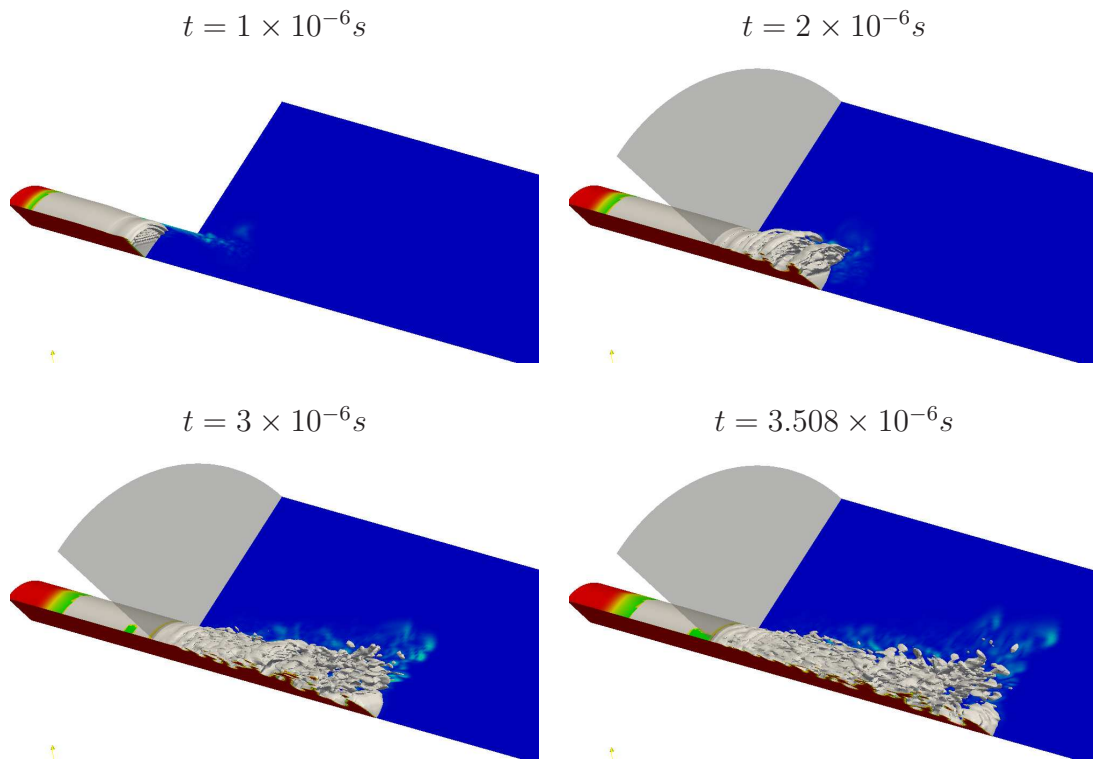
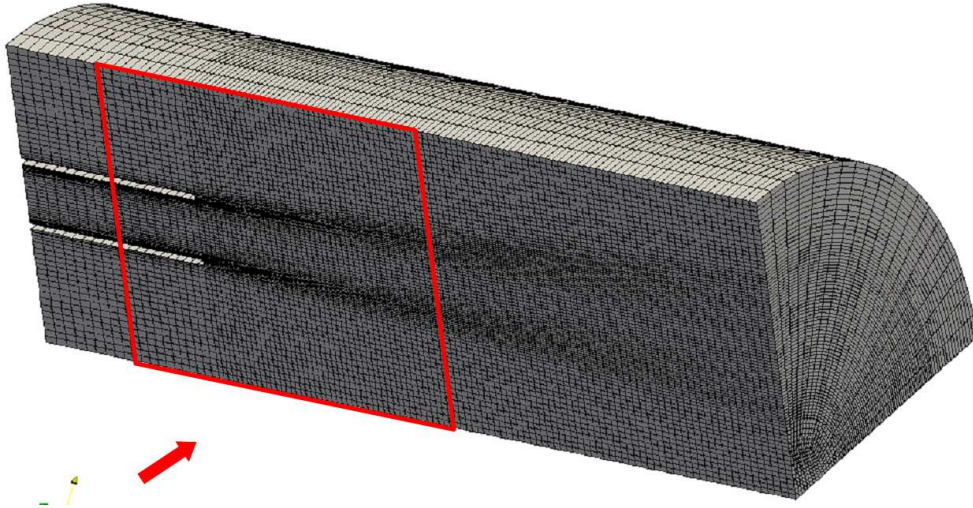


Figure 4.13: Simulation of Diesel jet breakup with OpenFOAM — Interface location, volume fractions and vorticity contours for the onset of the breakup

Isometric view of the domain — 1.86×10^5 cells



Detail of the nozzle outlet region

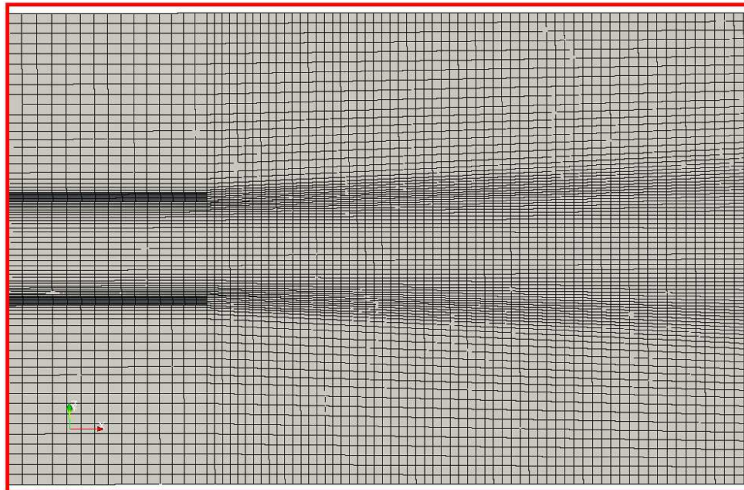


Figure 4.14: Simulation of axisymmetric sheet breakup with OpenFOAM — Mesh of the domain

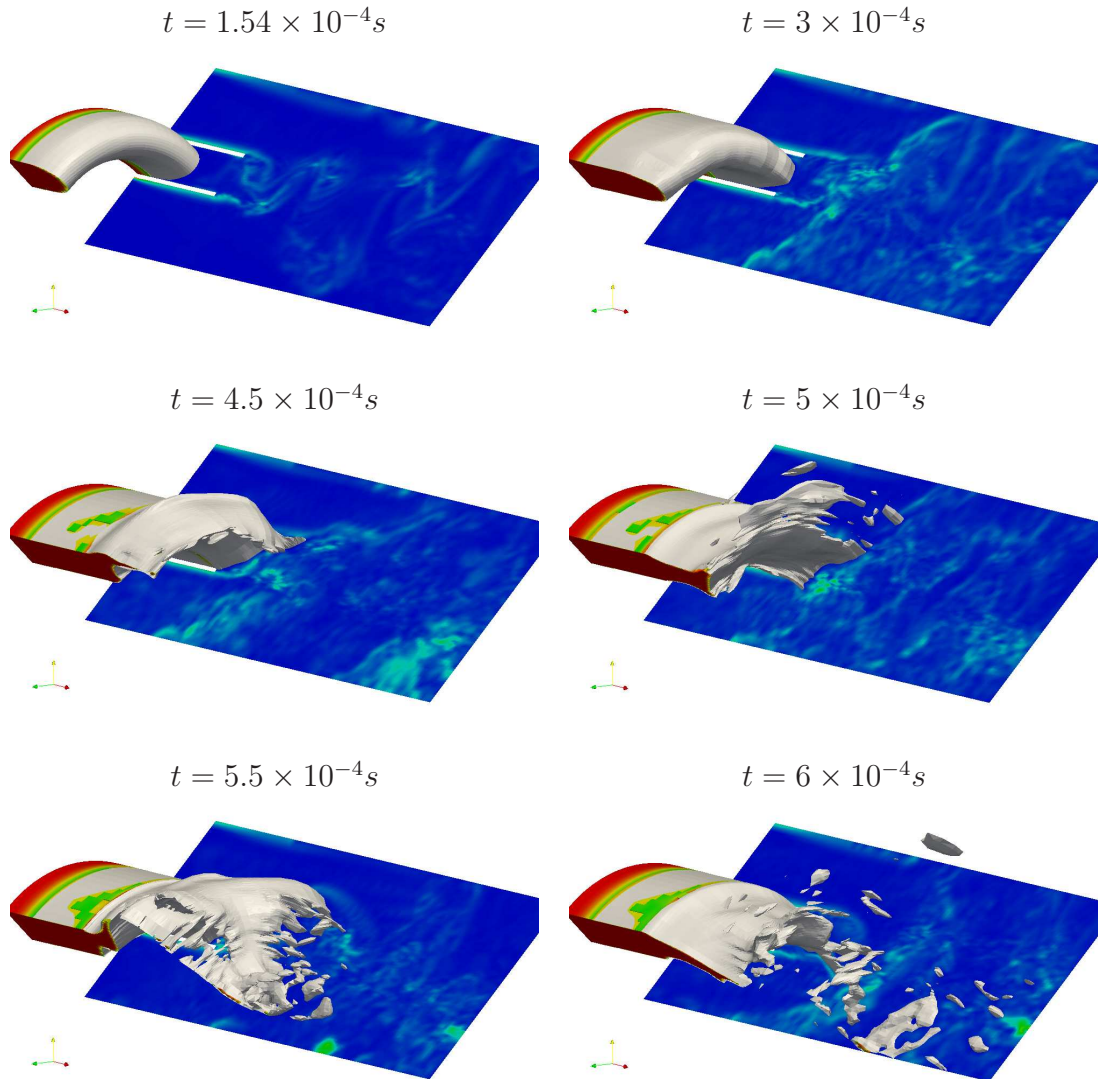


Figure 4.15: Simulation of axisymmetric sheet breakup with OpenFOAM — Interface location, volume fractions and vorticity contours for the onset of the breakup

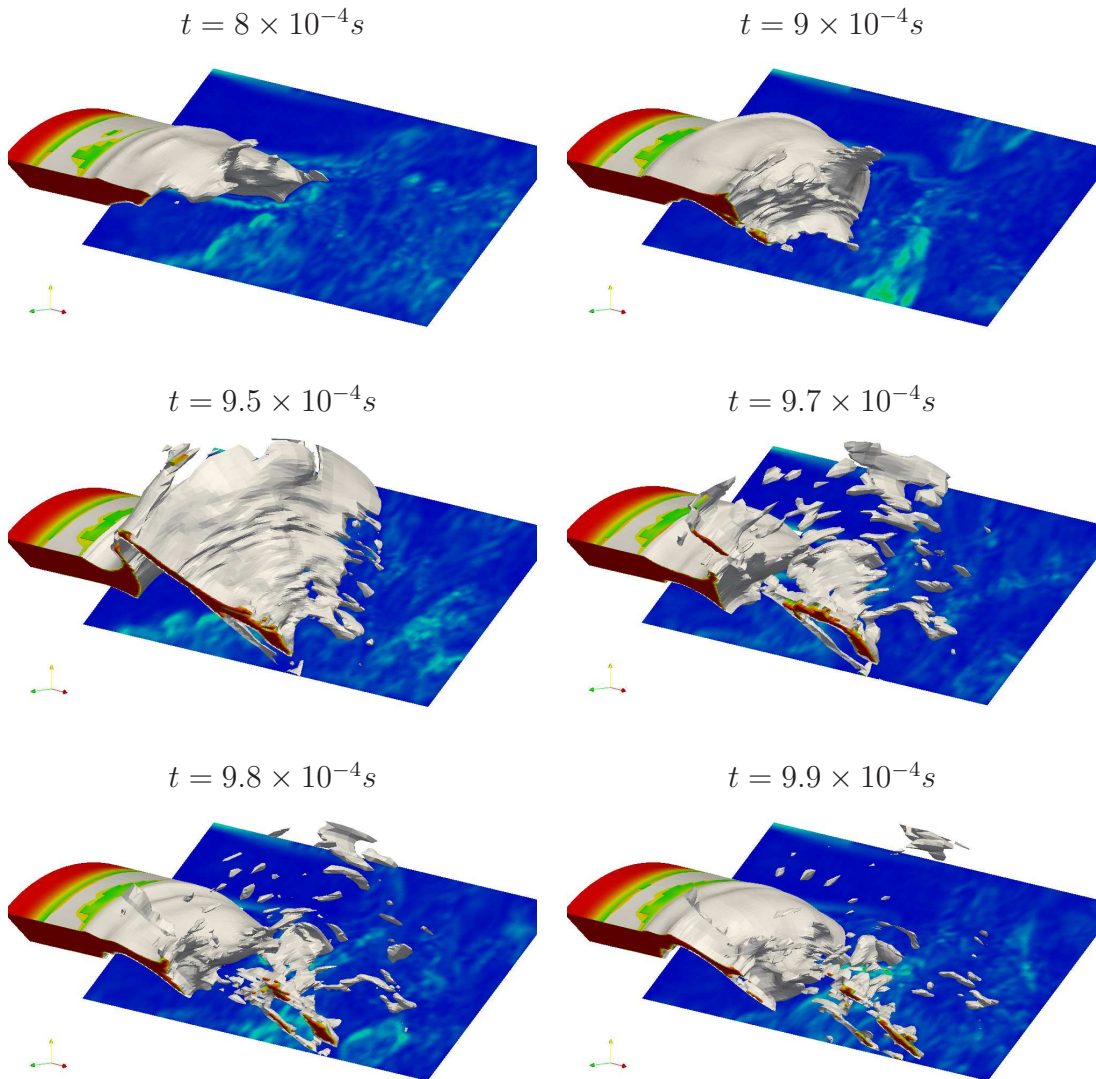


Figure 4.16: Simulation of axisymmetric sheet breakup with OpenFOAM — Interface location, volume fractions and vorticity contours for the breakup mechanism

A

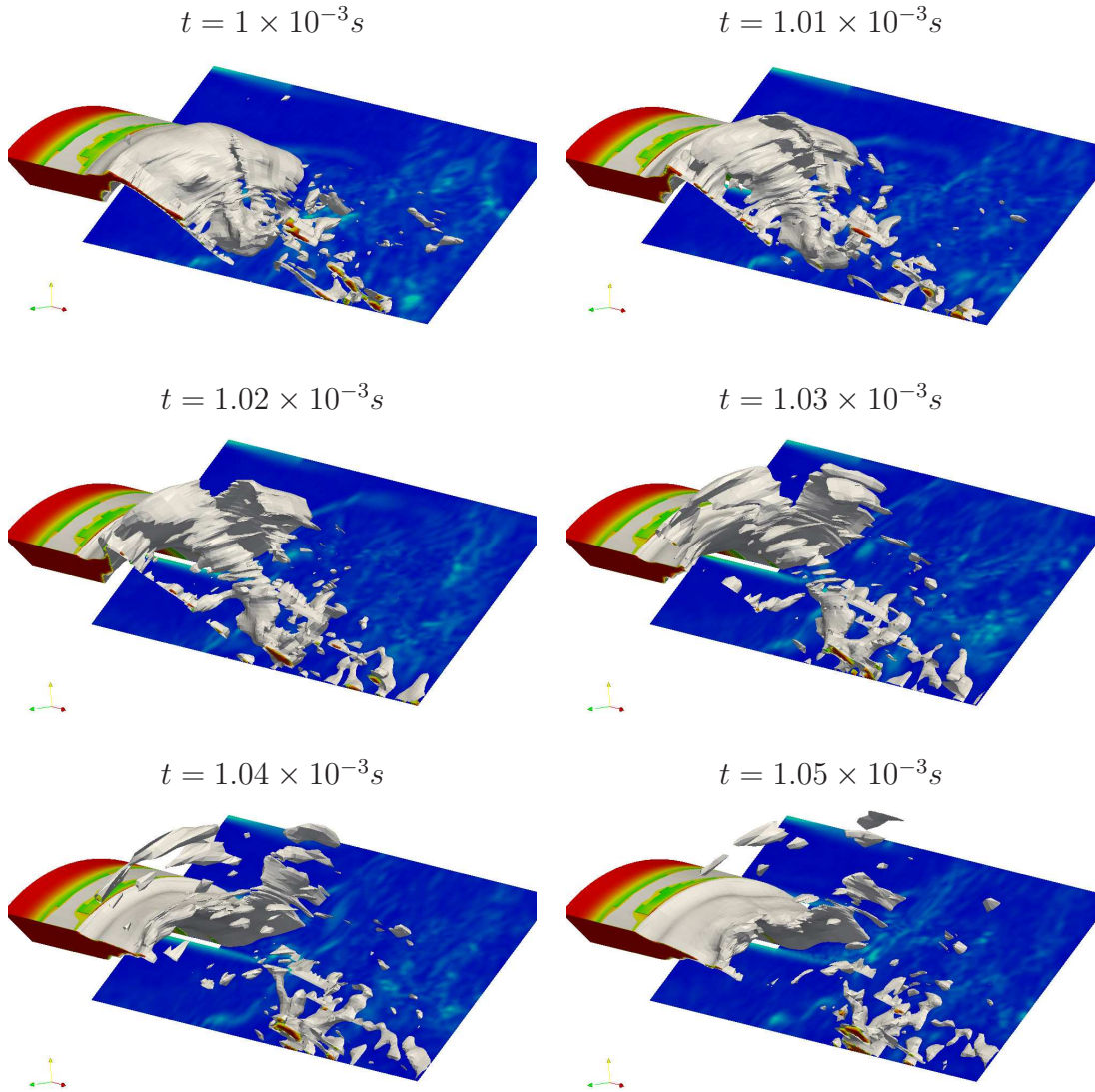


Figure 4.17: Simulation of axisymmetric sheet breakup with OpenFOAM — Interface location, volume fractions and vorticity contours for the breakup mechanism B

4.3.2 Simulations of sheet breakup without SGS model

Three calculations of flat sheet breakup are presented in this section, corresponding to three different gas velocities and therefore three different Weber numbers. All of these simulations have been run on the same geometry and with the same initial mesh (see Figure 4.18, top part). The geometry modelled is derived from the simulation of the axisymmetric sheet breakup and includes separator plates of finite thickness as recommended in [65].

The mesh is statically refined in the region where the liquid phase is expected in order to capture finer liquid structures. The resulting grid is then made up of cells of two different sizes: $\Delta x = 7.8\mu m$ in central part of the domain, and $\Delta x = 15.6\mu m$ elsewhere. During the calculation, it can happen that a small liquid structure, originally captured in the refined part of the grid, reaches the coarse part of the grid. Conveniently, when using statically refined mesh in Gerris, in such instance the code will automatically refine the mesh around the liquid structure in order to maintain its initial level of resolution (see Figure 4.18, bottom part). In the case of the simulation of atomisation at high We , this generally results in the rise of the cell count as the calculation time goes by.

Although very powerful for the calculation of multiphase flows, Gerris does not include sub-grid scale models. Therefore the validity of any simulation with Gerris (as it is currently), relies upon sufficient grid refinement to capture all the scales of the multiphase flow (turbulence and liquid structures). Table 4.13 provides the smallest scales of the multiphase flow: the Kolmogorov scale for both the liquid and the gas phase, and the droplet diameter obtained at $We = 10$, when the surface tension effects become important.

It is clear from this table that even at the lowest Weber number, none of the Kolmogorov scales are captured by the calculation. This is also true for the smallest liquid structures requiring at least two to five grid cells for a reasonable resolution. In particular, for the simulation at the lowest Weber number ($We = 410$), the finest

grid size should be quartered to capture the $d_{We=10}$ liquid structure. This further emphasises the need for sub-grid scale modelling in multiphase flows.

Even though all the scales are not resolved, some qualitative results may still be valid for these simulations, as several authors have successfully extracted useful information from calculations run at a similar level of resolution. These calculations may be perceived as some sort of “implicit large eddy simulations” (ILES), where the sub-grid contributions reduce to numerical diffusion (see [72] for a review of the method).

Table 4.13: Smallest scales of the flow field for all simulations without sub-grid scale model — Assuming a turbulence intensity of 5% in the liquid and 10% in the gas

| Simulation | $We = 410$ | $We = 2100$ | $We = 9300$ |
|---|-------------|-------------|-------------|
| Liquid phase Kolmogorov scale, η_{liq} | $0.52\mu m$ | $0.52\mu m$ | $0.52\mu m$ |
| Gas phase Kolmogorov scale, η_{gas} | $0.42\mu m$ | $0.25\mu m$ | $0.15\mu m$ |
| Droplet diameter at $We = 10$, $d_{We=10}$ | $4.9\mu m$ | $0.96\mu m$ | $0.22\mu m$ |
| Maximum mesh size, $\Delta x_{l=6}$ | | $15.6\mu m$ | |
| Minimum mesh size, $\Delta x_{l=7}$ | | $7.8\mu m$ | |

Sheet breakup at $We = 9300$ The evolution of the interface in the early stages of the fuel injection is presented in this paragraph for the sheet breakup at $We = 9300$. The growth of instabilities on the interface before the liquid leaves the channel is presented in Figure 4.19, and the onset of the breakup is shown in Figure 4.20

The interface starts to undergo some instabilities quite early in the injection channel (clearly observed at $t = 4 \times 10^{-5}s$). These waves, deforming the surface in both vertical and horizontal directions, grow rapidly to the point of generating small

pockets of gas near the separator plates, on either side on the horizontal symmetry plane ($t = 4 \times 10^{-5}s$).

At $t = 1 \times 10^{-4}s$ the instabilities have deformed the interface so strongly that some wave peaks announce the birth of ligaments, especially in the vicinity of the small pockets of gas where the surface is the most distorted. At this time step, the interface is already strongly asymmetric.

From $t = 1 \times 10^{-4}s$ onwards, droplets start to form: this is the onset of the breakup. The undulations start to produce distinctive “cells” at the interface with sharper boundaries. The liquid phase covers parts of the separator plates, and from the edges of these plates, at the location of the undulation peaks, thin ligaments are growing and shedding droplets of similar diameter.

All through these early stages of the breakup, the vorticity contours present patterns resembling Von Karman streets. These patterns of vorticity contours get more and more complex as the calculation time goes by, but maintain roughly the same large scale structures.

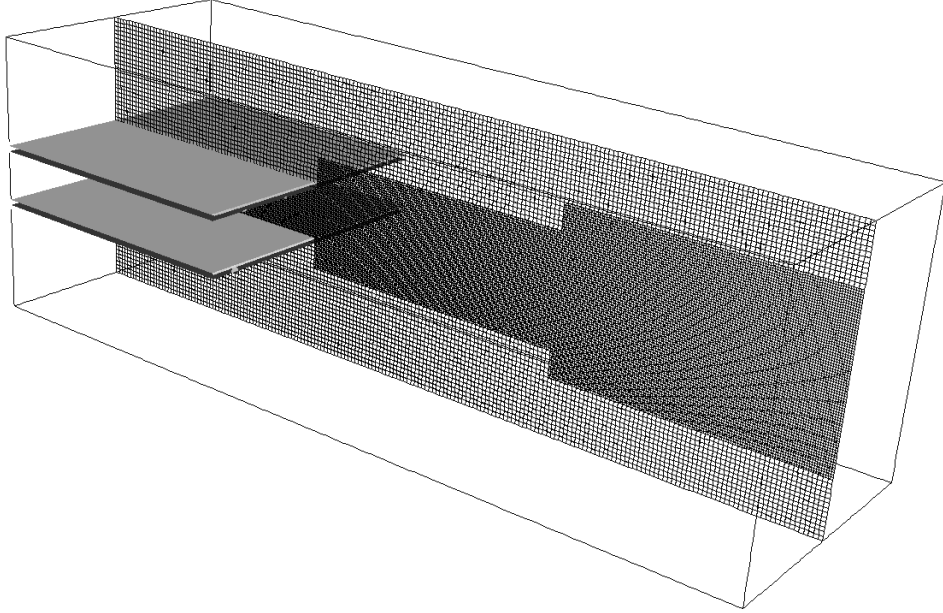
Effect of the We on sheet breakup The effect of the Weber number on the sheet breakup has been illustrated with three simulations run at increasing gas velocities, and therefore increasing We . The different breakup configurations obtained are shown in Figure 4.21, Figure 4.22 and Figure 4.23.

At low Weber number, $We = 410$, the liquid bulk extends downstream of the injection point and relatively thick ligaments are produced, leading to the breakup into large droplets. When the Weber number is increased to 2100, the interface adopts a totally different topology characterised by a shorter liquid bulk. At such conditions, the ligaments and droplets produced are much smaller and spread further outwards from the injection plane leading to a wider spray angle. When increasing the Weber number to 9300, similar trends are observed with the shortening of the liquid bulk and the decrease of the mean ligament thickness and droplet diameter.

It is worth mentioning that all three simulations of atomisation eventually lead

to the formation of non-physical cylindrical liquid structures in the middle of the computational domain. This further emphasise the fact that Gerris' development is still ongoing and that this code is aimed at academic research rather than component design.

Domain at $t = 0s$ — 2.96×10^6 cells



Domain at $t = 1.24 \times 10^{-4}s$ for $We = 9300$ — 3.51×10^6 cells

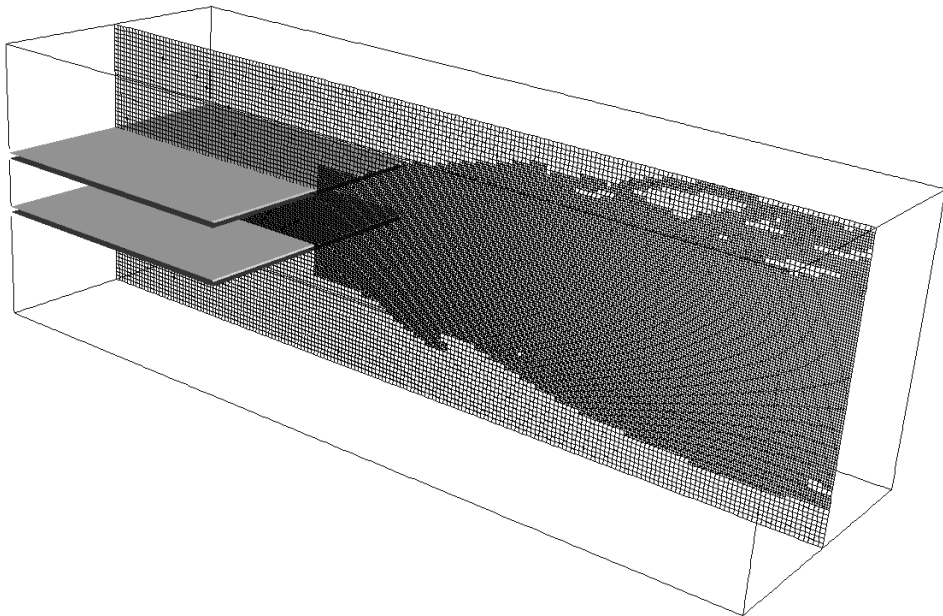


Figure 4.18: Simulation of flat sheet breakup with Gerris — Mesh of the domain

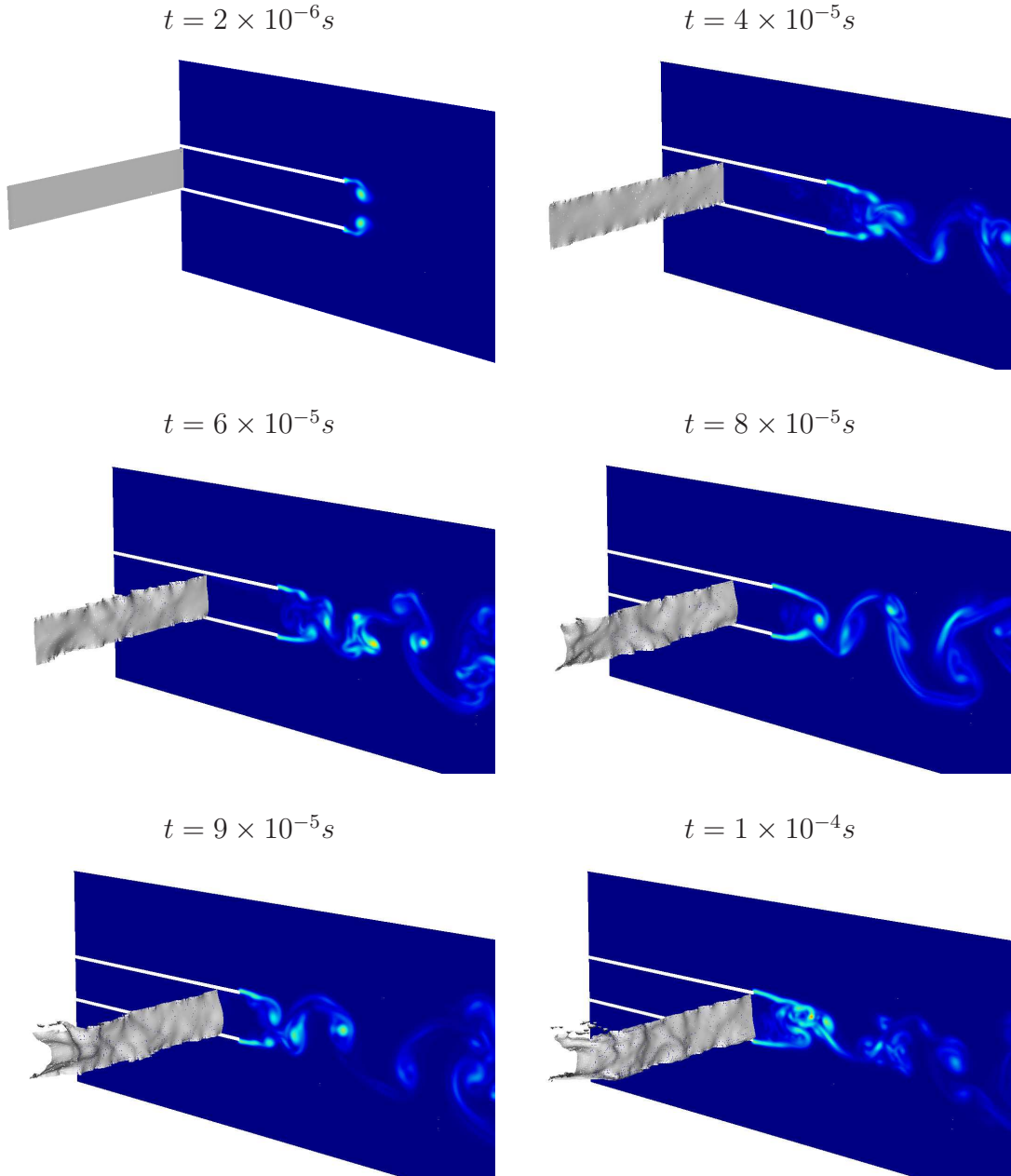


Figure 4.19: Simulation of flat sheet breakup with Gerris — Interface location and vorticity contours for the build-up of instabilities at the interface at $We = 9300$

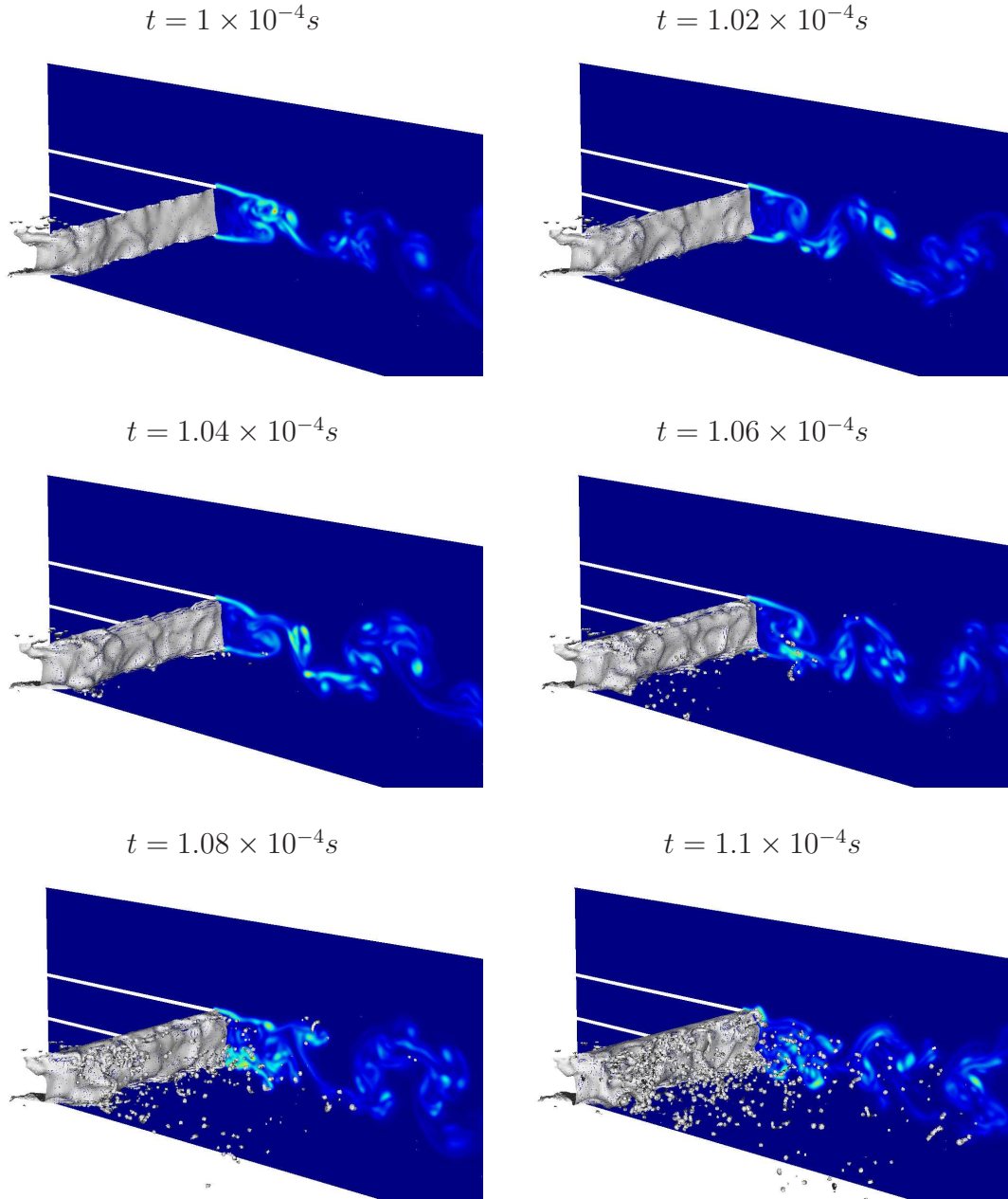


Figure 4.20: Simulation of flat sheet breakup with Gerris — Interface location and vorticity contours for the onset of the breakup at $We = 9300$

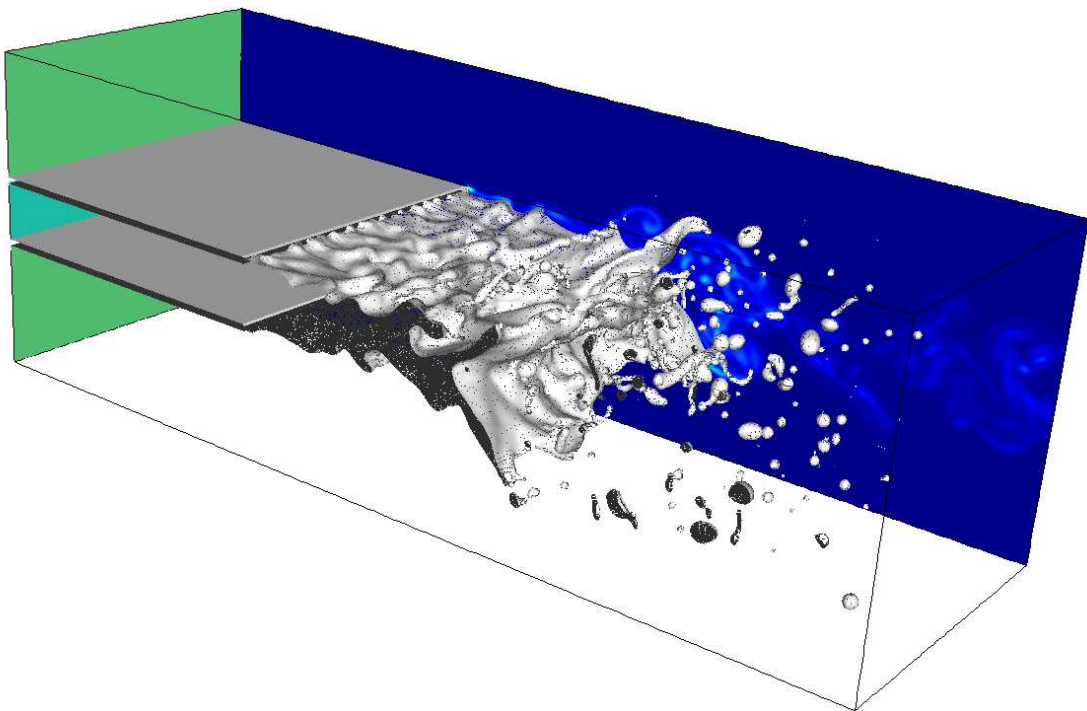


Figure 4.21: Simulation of flat sheet breakup with Gerris — Interface location, velocity magnitude and vorticity contours at $We = 410$ ($t = 1.78 \times 10^{-4} s$)

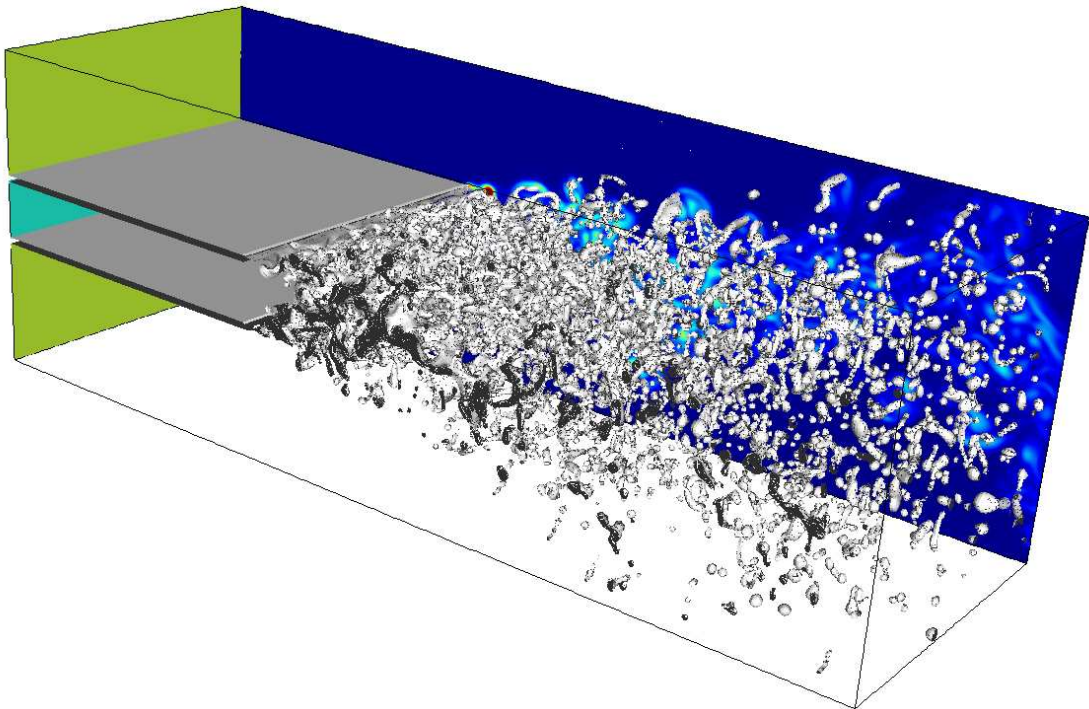


Figure 4.22: Simulation of flat sheet breakup with Gerris — Interface location, velocity magnitude and vorticity contours at $We = 2100$ ($t = 1.54 \times 10^{-4}s$)

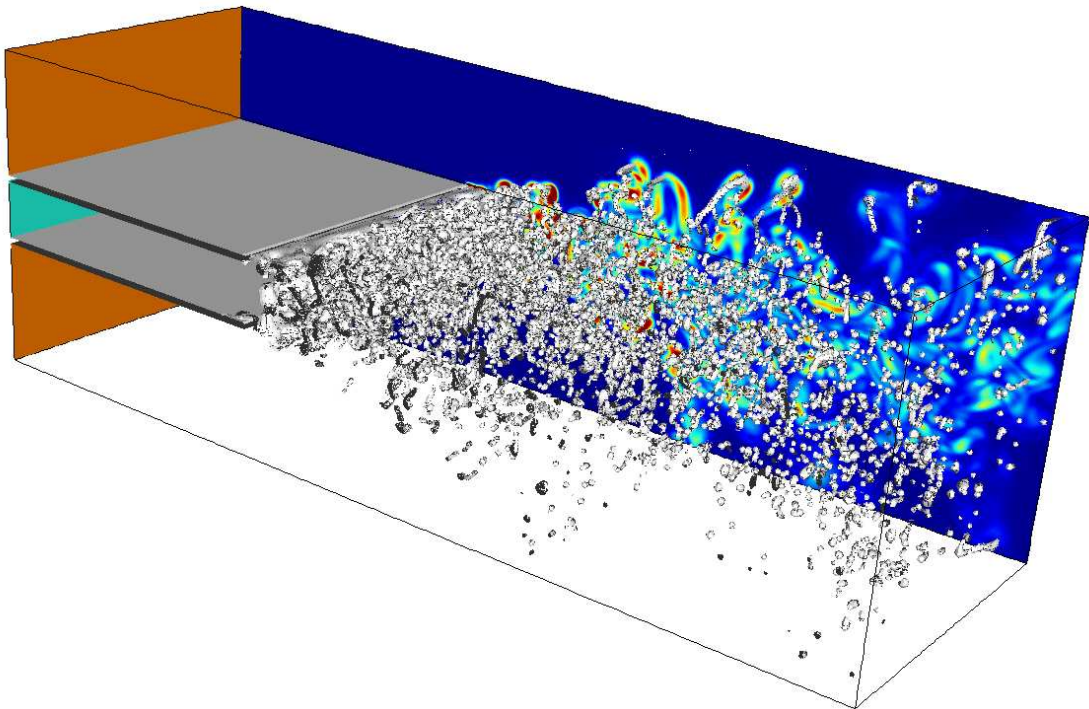


Figure 4.23: Simulation of flat sheet breakup with Gerris — Interface location, velocity magnitude and vorticity contours at $We = 9300$ ($t = 1.24 \times 10^{-4} s$)

Chapter 5

Atomisation modelling

The design of injectors relies heavily on testing as no numerical method has so far been able to provide validated quantitative results on a real air-blast atomiser geometry. However, in the past years this problem has attracted the interest of many scientists who have developed a broad variety of approaches to come closer to simulating the atomisation process and understand real injection systems.

The main successful attempts to model the atomisation process on simplified configurations are presented in Section 5.1. Then, in Section 5.2, we review the current approaches adopted by scientists to extend their modelling capability to the simulation of real engine geometries. Finally, in Section 5.3, we detail our strategy to simulate the fuel injection in a combustion chamber.

5.1 Demonstrated numerical capabilities

5.1.1 LES with VOF

The LES-VOF approach of Villiers et al. [44] has been applied to Diesel jet breakup using OpenFOAM. Even though the droplet size distribution obtained showed a strong grid dependency, the results of the simulations compared successfully with spray angle and liquid core length measurements. In this study of the atomisation

with LES, the authors primarily assumed that the unresolved eddies, surface perturbations and droplets did not affect the onset of the breakup. In addition, the method neglects the sub-grid scale effects of the surface tension and the interface-turbulence interaction.

Despite the poor VOF method employed (see Chapter 4), the calculations picked up the non-linear growth of Kelvin-Helmholtz instabilities along the jet surface but did not reproduce the mushroom shape (see Figure 5.1) typical of Diesel jets [150, 152]. The study also highlighted the sensitivity of the onset of the breakup to the turbulence generated in the nozzle upstream of the injection point. In addition, the comparison of a 90° sector simulation with full 3D calculations emphasised the three-dimensionality of the primary breakup. Sector models lead to the artificial limitation of the large scales of motion resulting in erroneous predictions of breakup length and spray angle.

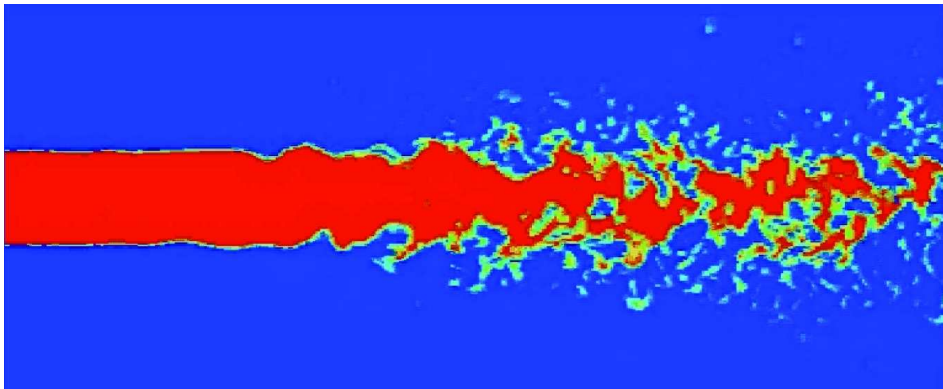


Figure 5.1: LES-VOF simulation of a Diesel spray atomisation (full 3D, $\Delta x_{min} = 10\mu m$, pipe $L/D > 40$) [44] — Contour of volume fraction

In their study, Villiers et al. assessed the degree of atomisation with the mean surface density, which increases with the level of interface distortion and with the concentration of small droplets. The analysis of the atomisation (droplet sizes and shapes) implied that breakup calculations reproduced the relative importance of surface tension, aerodynamic forces and inertia according to the scale considered.

In particular, the dominant influence of surface tension at small scales was illustrated by the spherical shape of small droplets whereas the predominance of aerodynamic forces and inertia at large scales was suggested by the elongated shape of the large liquid structures.

The effect of turbulence in the injection channel was also investigated qualitatively by Bianchi et al. using an LES-VOF approach [19]. Further simulations of Bianchi et al. [18] on a finer mesh ($\Delta x = cst = 4\mu m$) produced quantitative information about the effect of the nozzle regime on the atomisation process (see Figure 5.2). In particular, the authors noted that this phenomenon affected strongly the core length and the atomised mass but had almost no impact on the droplet size distribution because of the very high Weber number We , and Ohnesorge number Oh , of the liquid jet.

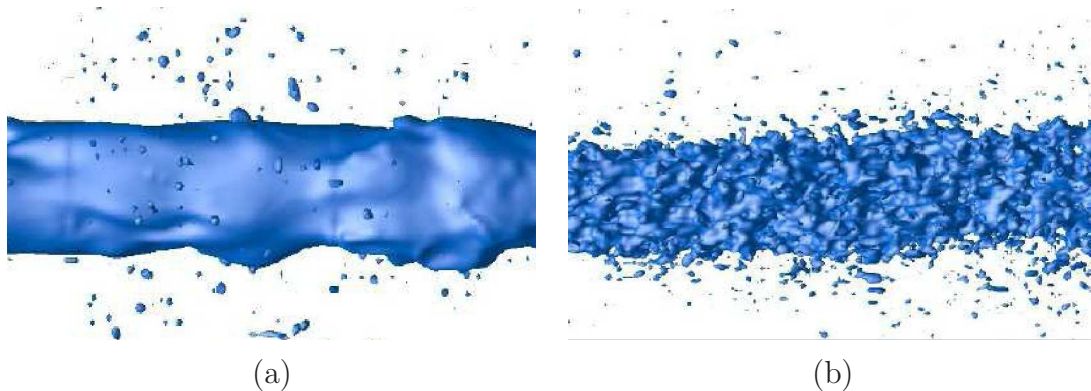


Figure 5.2: LES-VOF simulation of a Diesel spray atomisation ($\Delta x = 4\mu m$) — Effect of nozzle regime on the liquid surface perturbations at the injection point [18]: (a) $Re_{nozzle} = 1625$; (b) $Re_{nozzle} = 8725$

5.1.2 VOF with adaptive mesh refinement

Fuster et al. ran the VOF-AMR code Gerris [183] to simulate a hollow-cone atomiser at realistic automotive injection conditions [65]. In this work, the vorticity and the gradient of the tracer variable were used as criteria for the mesh refinement.

At first, Fuster et al. validated the code's capability to capture the behaviour of instabilities against the predictions of the viscous linear theory. In particular, Gerris matched the theoretical predictions for the temporal growth of disturbances within 5% for a broad range of conditions (parameters: density and viscosity ratios, surface tension).

Then, solving all the scales of the turbulence in both phases, the authors studied the onset of the instabilities, on a jet in co-flow, assuming that the unresolved liquid structures do not affect the appearance of waves on the interface. The high Re simulations, run in 2D (complex flow field, high resolution: $\Delta x_{min} = 0.8\mu m$), relied on a random vertical velocity (in the gas or the liquid phase) to trigger the instabilities; therefore avoiding the need to introduce any geometrical parameters in the analysis. In contrast, the low Re simulations, run in 3D, relied only on the modelled separator plate and the difference in inlet velocities to destabilise the interface.

The high Re calculations pointed out that the onset of the instabilities is strongly related to the turbulence at the inlet of both phases and that the amplification of the waves leads to the transition into the non-linear regime. These simulations also highlighted the sensitivity of the modes to the evolution of the bulk flow downstream (changes in velocity profiles, thickening of the boundary layer). Although good agreement was obtained with the theory in the linear part of the jet, the three-dimensionality of the flow — essentially downstream of the transition into the non-linear regime — suggested that 2D calculations were unsuitable for regions of the flow in the non-linear regime.

The low Re cases, run in 3D, captured the transverse instabilities that appear further downstream of the injection point. These simulations confirmed the two-dimensionality of the jet near the inlet and therefore tended to validate the 2D approach at the beginning of the linear regime region (before the growth of transverse instabilities). The addition of some injector geometry to the computational domain demonstrated the influence of both the thickness of the separating plate and the

vortices produced in the injection channels. In particular, these vortices appeared to affect significantly the primary breakup process and interacted strongly with the droplets generated.

Finally, Fuster et al. conducted a parametric CFD study of swirl-atomisers, investigated the sensitivity of the solution under grid refinement and validated their calculations ($(\rho_{liq}/\rho_{gas})_{CFD} = 30/1$) against experimental results ($(\rho_{liq}/\rho_{gas})_{EXP} = 700/5$). The numerical results appeared to be very sensitive to mesh refinement (see Figure 5.3) for both the small structures of the flow (droplets) and the big features of the spray (spray angle, breakup length). However, despite a significant difference in density ratio, the high resolution simulation ($\Delta x_{min} = 9\mu m$) reproduced the general features of the spray (cone angle, change of slope, breakup length) and the droplet size distribution. Because of the breadth of scales involved in the atomisation process, the simulation of the whole system is required to understand the operation of real injectors. This has been made possible — for a simple fuel injector — thanks to the use of adaptive meshing techniques that handle these issues efficiently.

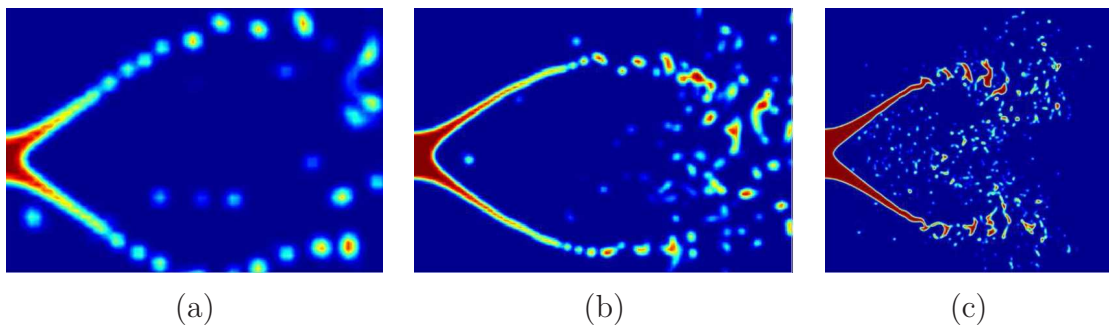


Figure 5.3: VOF simulation of a swirl outward-opening jet (full 3D) [65] — Contour of volume fraction in a median for various mesh sizes: (a) $\Delta x_{min} = 56\mu m$; (b) $\Delta x_{min} = 28\mu m$; (c) $\Delta x_{min} = 9\mu m$

5.1.3 Coupled LS-VOF combined with GFM

Menard et al. [150, 152] performed a CLSVOF simulation of a Diesel jet atomisation in the second wind-induced regime (see Chapter 2) using the GFM to handle the singularities. This method is particularly suited for the simulation of atomisation, as it minimises the mass loss thanks to the “VOF correction”, retains an accurate description of the interface geometry, due to the level set formulation, and applies accurately the jump conditions on a sharp interface with the GFM.

Menard et al. did not embed any sub-grid scale modelling in their framework, therefore aiming for DNS. The true DNS assumptions were questionable in both [152] ($\Delta x = 2.34\mu m$) and [150], where the most refined mesh ($\Delta x = 1.17\mu m$) was closer to resolve the smallest scales of both the turbulence ($\Delta x/\eta \approx 1.5$) and the liquid structures ($\Delta x/D_{min} \approx 0.5$ at $We = 10$).

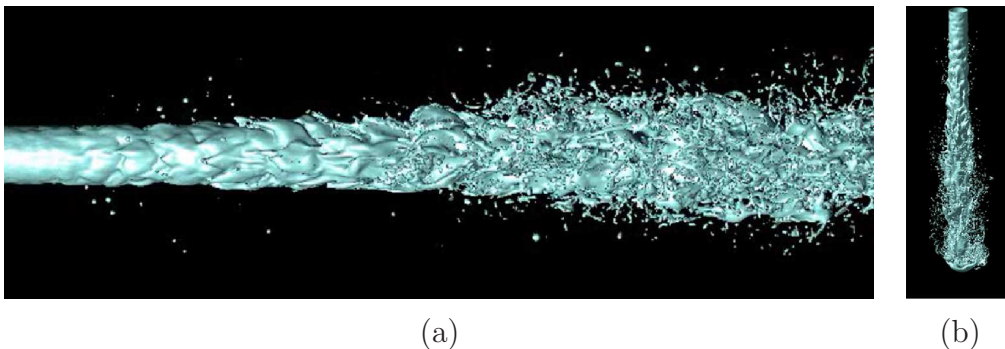


Figure 5.4: CLSVOF/GFM simulation of a Diesel spray atomisation [152] — (a) Liquid-gas interface near the injection point; (b) Developing spray

In their numerical framework, Menard et al. avoided simulating part of the injection system by generating turbulent inflow boundary conditions with the method of Klein et al. [112], which produces correlated random velocities based on a given length scale. The simulations of the Diesel jet reproduced satisfactorily the breakup length, the spray angle, the mushroom shape of the tip of the jet and the 3D instabilities near the injection point (see Figure 5.4).

Nevertheless, in their analysis of the evolution of the droplet distribution under grid refinement, the authors noted that the insufficient level of resolution affected their calculations. In particular, they concluded that the coarseness of the grid resulted in the predominant occurrence of a “numerical” breakup process for both droplets and thin ligaments. However, on instances of physical breakup, these calculations matched the prediction of droplet sizes as a function of ligament thickness given in [146].

5.1.4 RLSG combined with Lagrangian tracking

Kim et al. have combined the RLSG-balance force method of Herrmann [86] with the Lagrangian tracking of small droplets to produce near DNS of a liquid jet in co-flow [111]. The settings of the simulation match the experimental conditions of [146]. Kim et al. introduced a novel procedure to take small spherical liquid structures (i.e. no ligaments) out of the level set formulation in order to transfer them into a “Lagrangian stochastic spray” model.

The transfer of droplets into a Lagrangian spray model saves computational time while retaining an accurate description of the liquid phase. Indeed, the atomisation process produces a large amount of droplets and describing all the liquid structures with a level set formulation becomes very expensive in the early stages of the development of the jet. In addition, Lagrangian spray models have demonstrated satisfactory performance in the representation of the secondary breakup. However, in [111], the transfer of droplets is only based on the size and the shape of the particle and does not account for the frequent occurrence of droplets re-entering the bulk of the liquid [1].

The calculations showed qualitative agreement with the experimental results of [146]. In particular, it reproduced the series of instabilities (longitudinal Kelvin-Helmoltz followed by azimuthal Rayleigh-Taylor) leading to the formation of ligaments. The ligament’s stretching, pinch off and breakup under Plateau-Rayleigh

instability were also satisfactorily reproduced (see Figure 5.5). However, as in [44, 150, 152], the droplet size distribution appeared grid dependent and the peak of the droplet size distribution was also obtained for a droplet diameter of twice the minimum grid size. Kim et al. still claimed some level of grid-convergence of the droplet size distribution for droplets larger than two G-grid cells (finest grid size in the RLSG framework).

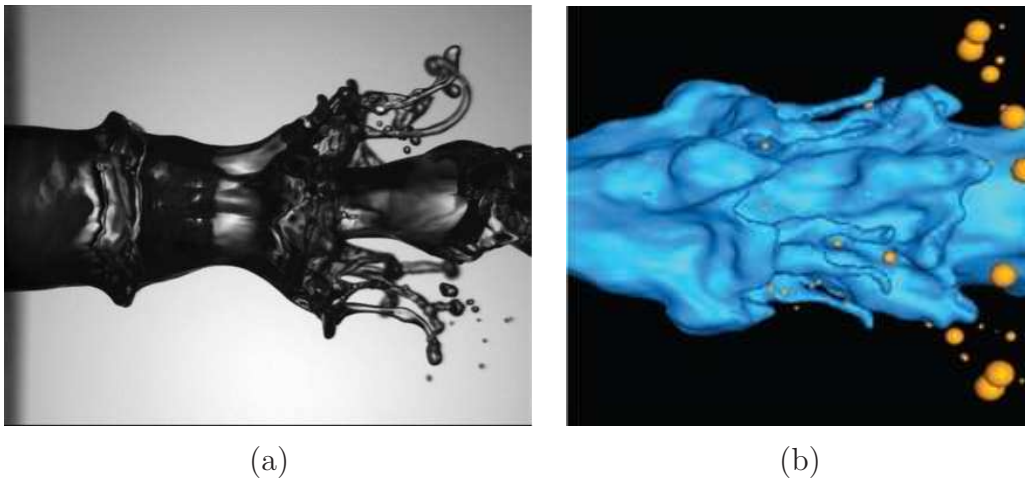


Figure 5.5: Round liquid jet sheared by co-flowing air ($Re_{gas} = 3770$; $Re_{liq} = 295$) — (a) Experiments [146]; (b) RLSG/Lagrangian tracking simulation [111]

5.1.5 Conservative LS with GFM

Desjardins et al. modelled Diesel jet atomisation with an improved CLS method combined with the GFM to treat the singularities [47]. This method maintains satisfactory mass conservation — required for the stability of the calculation and the accuracy of the results — at a much lower computational cost than the CLSVOF method of Menard et al. [150, 152]. Desjardins et al. did not include any sub-grid scale models, therefore aiming for DNS with a constant grid size of $\Delta x = 3.9\mu m$. However, Desjardins et al. did not extract any quantitative information from their simulation, conscious that true DNS would not be reached with such resolution.

The simulation domain and conditions were similar to [152] but did not include any turbulent inflow conditions nor any modelling of the injection channel. Although this questions the realism of this simulation, as most authors [18, 19, 44, 146] acknowledged the importance of this factor, it facilitated the study of the induction of turbulence in the gas phase by the resulting jet. In particular, Desjardins et al. highlighted annular vortical structures generated by the jet in the gas and reported their strong interactions with liquid structures.

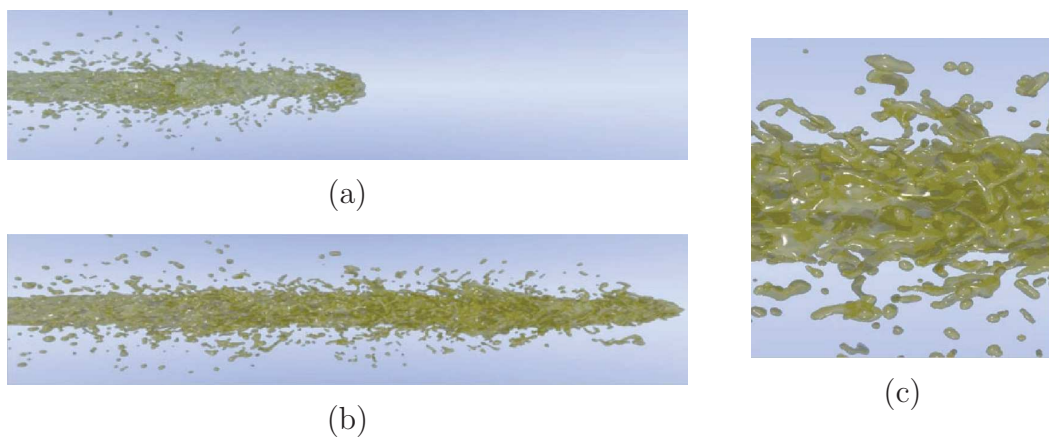


Figure 5.6: CLS/GFM of Diesel spray atomisation [47] — (a) “Mushroom shape” of the jet tip shedding droplets; (b) Ligament formation and breakup; (c) Detail of the liquid structures

5.2 Towards an industry-friendly approach

5.2.1 Development of sub-grid scale models

Most importantly, the main authors have realised the need to develop models for the sub-grid contribution of the interface (in particular for LES of atomisation). As the thickness of the interface is of the order of a few tenths of nanometres, the resolution requirements for representing the smallest liquid structure in the case of a topology change (“zero-sized connecting ligament at the very instance of the breakup” [71])

would be unmanageable in the simulation of turbulent primary breakup; let alone the violation of the continuum assumption of the Navier-Stokes equations. Therefore, the generation of sub-grid scale models (droplet pinch off, interface-turbulence interaction, interfacial sub-grid contributions, ligament breakup, etc . . .) is essential to produce numerical simulation of atomisation that carries a physical significance. For this purpose, various authors are developing specific numerical methods that embed a physical description of the interface [84, 118]. In this context, Van der Waals-Cahn-Hilliard frameworks, such as the phase field methods [101], also seem promising.

5.2.2 Realistic boundary conditions

Many authors have observed — both numerically and experimentally — the importance of the boundary conditions (inflow and geometry) on the liquid jet atomisation and have identified the need to reproduce them accurately. The accurate application of both the velocity profiles (boundary layers in liquid and gas phases — [146]) and the inlet turbulence [18, 44, 54] are critical to predict the onset of the breakup and the growth of perturbations at the interface. In particular, the inlet turbulent conditions have been prescribed in two different ways: some authors have simulated the flow in the injector channels [18, 44] whereas others have developed specific procedures [112, 150, 152] to avoid the extra computational cost of an extended domain.

As pointed out by Fuster et al. [65], the understanding of real injection systems requires the resolution of large scale turbulent structures as these eddies interact with themselves and with the liquid particles. As a result, these vortices, resulting from the geometry of the whole system, strongly influence the overall structure of the spray and the final droplet distribution. This is further confirmed by the high level of accuracy obtained by [6, 7, 67, 200] in the LES of atomising sprays with stochastic modelling of secondary breakup.

In order to embed such a large amount of geometry in the numerical calculation, the use of adaptive mesh refinement is an efficient way to deal with the breadth of scales involved in the atomisation process [86, 182]. It is also necessary to optimise, the use of computational power. For this purpose, Peng et al. developed a “narrow-band” approach, for level set methods, that localises the resolution of the equations of the interface-capturing method [173]. With the same aim of reducing the computational cost, Kim et al. transferred the small droplets from the interface-capturing technique to a Lagrangian spray model [111]. This technique can be further improved by implementing a criterion that accounts for the droplets re-entering the bulk of the liquid phase. More generally, the authors have concentrated their effort on developing highly parallelisable methods usually based on VOF and/or LS interface description. Also, some authors have focussed on the development of LES formulations for atomisation modelling. Such frameworks can include much more geometry than DNS, which is never truly achieved in high Re atomisation.

5.2.3 Accurate description of the interface

The internal geometry of aero-engine injectors is very complex and has a dominant influence on the spray pattern. In order to account for the effect of these geometrical details on the spray characteristics, the numerical frameworks have to embed a method to capture the interface. As none of the available methods have satisfactorily described the atomisation, the development of interface description methods is ongoing.

In the recent years, many authors have invested a lot of effort in the correction of the flaws of the existing methods. More precisely, in the case of VOF interface-capturing techniques, significant improvements have been made on the treatment of surface tension [59, 183] and the reconstruction and the advection of the interface (see [178] for a review). For level set methods, the mass conservation properties have been enhanced either by coupling with VOF (CLSVOF — [25, 150, 152, 243])

or, more cheaply, by modifying the scalar advected (CLS — [47, 163]).

The hybridising of existing interface description methods has been very popular in the past ten years. In particular, the LS, VOF and marker methods have been coupled — two by two — to combine the relative advantages of the different techniques. Although these approaches have usually given accurate results, they generally involve high computational cost [53, 133, 150, 152, 185]. In addition to that drawback, the coupling with surface marker particles is in general very difficult to extend to 3D and does not always represent physically the sub-grid scales of the hydrodynamic field [217].

The difficulty of developing efficient new ways of describing the interface and the complexity involved in the correction of the existing techniques have led many scientists to consider transforming obsolete or “exotic” methods developed in other fields of science. For instance, the smoothed particle hydrodynamics (SPH) of Gingold and Monaghan [68], originally developed for astrophysics calculations, seems to attract a lot of interest nowadays. Although these methods tackle the problem of primary breakup with a fresh perspective, they do not generally describe multiphase flow problems with the appropriate physics.

5.2.4 Numerical implementation of the physics

The challenge of handling the complex physics involved in the atomisation process is directly related to the treatment of the interface. The numerical description of material discontinuities and surface tension has been tackled with significant success in the past years. In particular, Fedkiw et al. have produced the ghost fluid method [55] — essentially directed towards LS formulations — that retains a sharp interface while accurately applying the jump conditions. Furthermore, the popular continuum surface force method has been significantly improved by François et al. who produced a “balanced force algorithm” [59] to maintain the exact balance between the surface tension forces and the pressure jump across the interface. These methods

have significantly reduced the occurrence of unphysical velocities (spurious currents) near the interface and the focus is on the development of accurate calculations of the interface curvature [86, 183, 198] to fully address this problem.

The addition of the compressibility effects in multiphase flows (e.g. [154] for VOF, [157] for LS) has been considered by few authors. Tanguy and Menard [150, 246] have noticed the potential need for a compressible formulation from code instabilities arising in the simulation of pockets of gas trapped between liquid membranes during the formation of ligaments. Considering the simplifications resulting from the incompressible formulation and taking into account the large amount of work invested in the development of dedicated numerical schemes, the generation of an efficient compressible multiphase solver remains also a challenge.

5.2.5 Summary

The main authors have pursued the development of the following fields to improve the simulation of atomisation:

- Generation of SGS models for LES of atomisation
- Development of hybrid interface description methods to maximise the advantages of the different techniques
- Correction of VOF and LS flaws
- Adaptive mesh strategies
- Simulation of primary breakup with a compressible formulation
- Lagrangian tracking of small droplets

More practically, the understanding of primary breakup and more generally the successful simulation of high Re atomisation is of particular relevance to the numerical calculation of the reacting flow in combustion chambers. Therefore, in order

to focus the generation of methods on the industrial problem, there is a need to assess the droplet size below which the liquid structures have negligible effects on the reacting flow.

Furthermore, with the advent of high pressure propulsion systems, the injection of fuel may occur at supercritical conditions. In such an environment, the surface tension becomes negligible and the concept of interface is questionable [71]. Just like the computation of primary breakup, the numerical simulation of “supercritical fuel injection” is an open field of research with strong implications for both the industry and the environment.

5.3 A modelling capability for fuel-injector design

Now that the state-of-the-art and the current trends in atomisation modelling have been presented, this section provides the structure of the CFD capability developed to simulate the atomisation process in aero-engine combustors.

The requirements of the industry are listed in Section 5.3.1 and the strategy adopted to meet these objectives is detailed in Section 5.3.2. Then, the structure of the resulting modelling tool is outlined in Section 5.3.3.

5.3.1 Industry requirements

To satisfy the industry’s need for a modelling capability simulating fuel-injection, this project has focussed on the following criteria to develop a novel multiphase flow solver:

- Handling of the complex geometries of modern aero-engines fuel injectors.
- Fast computations to provide results within engineering time scales.
- Accurate calculations to capture the physics of the atomisation.
- Mass conservation of the liquid fuel.

- Robust numerical formulation to ensure simulation convergence for all operating conditions.
- Production of data ready to be exploited by combustion codes.
- Parallelised solver to make the most of the available computational power.

5.3.2 Modelling strategy

In order to represent faithfully the geometry of the injection device in a timely manner, the method developed is geared towards general unstructured grids. Besides, as unstructured meshes are generally faster to generate, the use of such numerical methods allows to significantly reduce the mesh generation task of the CFD design study.

As CFD studies need to satisfy engineering time scales without compromising on the quality of the calculation, our approach maximises the amount of pre-processing operations. Once stored in text files, the mesh-dependent variables can be read directly before any new calculation on a given mesh. It results that the amount of run-time operations required is minimal and that high-order schemes run as fast as low-order ones. Since a given geometry will be tested at many different conditions with the same mesh, this pre-processing strategy makes the overall CFD study very efficient.

In addition, the mass conservation of the fuel present in the flow field is critical to accurate combustion predictions. As the solver developed can also be used as a sub-model for combustor design, the two-phase flow modelling is based on an improved conservative level set method that ensures mass conservation while remaining cheaper than accurate coupled level set-VOF.

The modelling capability has been implemented with a WENO scheme developed for unstructured meshes [186, 187]. In our numerical experiments, moving from the high-order linear to the WENO formulation of the scheme has proven to increase

dramatically the stability of the method in the case of very large density ratios (see Section 8.4). In particular, our modelling capability has demonstrated significantly more robustness than the low-order method of OpenFOAM on the falling droplet case of Sussman et al. [241].

The method outputs droplet fields and statistics that can be directly imported into Lagrange tracking solvers for combustion modelling purposes. The dataset produced by the multiphase modelling capability is also useful to analyse the performance of a given injector design. Once identified in the flow-field, the droplets are removed from the level set field and transferred into a droplet field. With this approach the atomisation process is described accurately at a lower computational cost.

Finally, a significant amount of effort has been placed in the parallel implementation of the method. It involved parallelising the unstructured numerical scheme, the novel conservative level set method and the droplet transfer.

5.3.3 Outline of the solver

As mentioned in the previous sub-section, the code involves both pre-processing and run-time operations. The pre-processing being only performed once on a given mesh. The pre-processing operations involve producing mesh dependant variables for:

- The high-order scheme developed for general unstructured meshes (see Chapter 6). Most of the pre-processing computational time is devoted to that task.
- The high-order gradient involved in the interface normal calculation (see Section 7.3).
- The transfer of droplets (see Section 8.2).
- The parallel run of the method (see Section 8.3).

For the run-time operations, the modelling capability developed presents the following structure:

1. Update of the physical properties of the multiphase flow-field according to the liquid volume fraction.
2. Transport of the liquid volume fraction represented by a conservative level set field.
3. Calculation of the surface tension term.
4. Update of the turbulence properties if using a LES framework.
5. Pressure-velocity coupling as described in [208]. The discretised momentum equation is first constructed and then a “correction loop” is performed. At each iteration of that loop the pressure equation is solved and the momentum is corrected based on the pressure change.
6. Transfer of droplets from the level set field to the “Drops” field and output of the discrete phase characteristics.

In the generation of this numerical tool, the main focus has been placed on the development of a novel high-order scheme for unstructured meshes (see Chapter 6) and the production of a robust conservative level set method (see Chapter 7). These essential building blocks of the new modelling capability produced have then been implemented in parallel with an algorithm to output droplet fields and characteristics (see Chapter 8).

Chapter 6

A high-order scheme for general unstructured meshes

In this chapter, we detail the methodology developed, and implemented in parallel, to construct high-order schemes — linear and WENO — on 3D mixed-element unstructured meshes, consisting of general convex polyhedra. This numerical scheme can formally reach arbitrarily high order of accuracy in space.

The construction of WENO schemes on 3D unstructured meshes is notoriously difficult, as it involves a much higher level of complexity than 2D approaches. This is due to the multiplicity of geometrical considerations introduced by the extra dimension, especially on mixed-element meshes. Therefore, we have specifically developed a number of algorithms to handle mixed-element meshes composed of convex polyhedra with convex polygonal faces.

The linear reconstruction procedure produced for 3D mixed-element unstructured grids is detailed in Section 6.3.1. Then, in Section 6.3.2, the main points of the WENO reconstruction are presented. In Section 6.3.2, the computation of the numerical flux is explained. The application of the technique to the solution of the level set equation is considered in Section 6.4. Results are presented in Section 6.6 for a set of test cases in 2D and 3D.

6.1 WENO schemes for unstructured meshes

WENO schemes were originally developed for Cartesian grids [107, 131]. They were an evolution of the essentially non-oscillatory (ENO) schemes introduced by Harten et al. in 1987 [77, 78] to achieve high-order accuracy and non-oscillatory properties near discontinuities such as shock waves in high-speed compressible flows. As the interest in unstructured meshes grew, WENO schemes were constructed for triangular [60, 95] and tetrahedral meshes [49, 50, 286]. However, the construction of such schemes is much more complicated on unstructured meshes than on Cartesian grids as no particular direction can be identified in the distribution of the elements.

The basic principles of the construction of WENO schemes for triangular meshes were presented by Friedrich [60], based on the work of Abgrall [4]. Later, Dumbser and Käser extended these ideas to tetrahedral meshes [49] and defined an approach to devise arbitrarily high-order schemes. Titarev et al. [254] improved this approach for two-dimensional (2D) computational domains and produced high-order schemes on mixed-element 2D meshes. We have extended the approach of Dumbser and Käser [49] and Titarev et al. [254] to 3D mixed-element unstructured grids for linear hyperbolic equations [186]. In [187], we presented an extension of our scheme [186] to general polyhedral cells and applied it to the solution of the level set equation and the Burgers' equation.

6.2 Overview of the numerical scheme

Looking ahead to the need to solve hyperbolic conservation laws to capture the interface (see Chapter 7), the design of our high-order accurate numerical scheme is based on two theorems: Godunov's theorem [70] and the Lax-Wendroff theorem [121]. The Lax-Wendroff theorem implies that if the hyperbolic conservation laws are solved with conservative numerical schemes, the conserved variable — in our case the smeared out liquid volume fraction (see Chapter 7) — should be conserved

exactly. As for Godunov's theorem, it suggests that it is necessary to resort to a non-linear numerical scheme (e.g. WENO) to reach a high order of accuracy without generating oscillations.

The scope of the method is restricted to general hyperbolic systems of first-order partial differential equations [49, 50]. Such systems may be expressed in 3D as:

$$\frac{\partial}{\partial t} \mathbf{U} + \frac{\partial}{\partial x} \mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y} \mathbf{G}(\mathbf{U}) + \frac{\partial}{\partial z} \mathbf{H}(\mathbf{U}) = \mathbf{0} \quad (6.1)$$

where \mathbf{U} is the vector of conserved variables and $\mathbf{F}(\mathbf{U})$, $\mathbf{G}(\mathbf{U})$ and $\mathbf{H}(\mathbf{U})$ are the vectors of the fluxes respectively in the x , y and z directions.

The computational domain is denoted by Ω and is discretised using conforming elements E_i of volume $|E_i|$ and boundary ∂E_i . Integrating (6.1) over the element E_i leads to:

$$\iiint_{E_i} \frac{\partial}{\partial t} \mathbf{U} \, dE_i + \iiint_{E_i} \nabla \cdot \mathcal{A} \, dE_i = \mathbf{0} \quad (6.2)$$

with the divergence of the second rank tensor $\mathcal{A} = (\mathbf{F}, \mathbf{G}, \mathbf{H})$ given by:

$$\nabla \cdot \mathcal{A} = \frac{\partial}{\partial x} \mathbf{F}(\mathbf{U}) + \frac{\partial}{\partial y} \mathbf{G}(\mathbf{U}) + \frac{\partial}{\partial z} \mathbf{H}(\mathbf{U})$$

Assuming that the control volume E_i is fixed and therefore independent of time t and introducing $\bar{\mathbf{U}}_i$, the cell average of the conserved variables at time t , we can re-write the first term on the left-hand-side (l.h.s.) of (6.2) as:

$$\iiint_{E_i} \frac{\partial}{\partial t} \mathbf{U} \, dE_i = \frac{d}{dt} \iiint_{E_i} \mathbf{U} \, dE_i = |E_i| \frac{d}{dt} \bar{\mathbf{U}}_i \quad (6.3)$$

Inserting (6.3) in (6.2), and applying the divergence theorem to the second term on the l.h.s. of (6.2), leads to the finite volume formulation:

$$\frac{d}{dt} \bar{\mathbf{U}}_i + \frac{1}{|E_i|} \iint_{\partial E_i} \mathcal{A} \cdot \mathbf{n} \, d(\partial E_i) = \mathbf{0} \quad (6.4)$$

where $\mathbf{n} = (n_x, n_y, n_z)$ is the outward unit vector normal to the surface ∂E_i and $\mathcal{A} \cdot \mathbf{n} \, d(\partial E_i)$ is the flux component normal to ∂E_i .

Splitting the integral over the contour of the element ∂E_i into L_i integrals over the faces F_l of E_i , and introducing the vector $\mathbf{A}_{n_l} = \mathcal{A} \cdot \mathbf{n}_l$ (\mathbf{n}_l being the outward unit vector normal to F_l), we express the second term on the l.h.s. of (6.4) as:

$$\iint_{\partial E_i} \mathcal{A} \cdot \mathbf{n} \, d(\partial E_i) = \sum_{l=1}^{L_i} \iint_{F_l} \mathbf{A}_{n_l} \, d(F_l) = \sum_{l=1}^{L_i} \mathbf{A}_{il} \quad (6.5)$$

The inter-cell flux \mathbf{A}_{il} associated with the face F_l of the element E_i is efficiently calculated with a Gauss-Legendre quadrature of appropriate order. Inserting (6.5) into (6.4) leads to:

$$\frac{d}{dt} \bar{\mathbf{U}}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \mathbf{A}_{il} = \mathbf{0} \quad (6.6)$$

In order to time-advance the cell-averages of the variables, the finite volume scheme (6.6) requires the determination of the inter-cell fluxes. A reconstruction operator is therefore necessary in each element E_i to provide the point-wise values of the solution as needed by the flux evaluation. By using a polynomial reconstruction operator, the finite volume method can reach arbitrarily high orders of accuracy in space on any type of grid. If coupled with the ADER method of Toro et al. [110, 221, 222, 252, 253, 256–258], this methodology would lead to a arbitrarily high-order numerical scheme in both space and time. In this work, we coupled the scheme with a Runge-Kutta time scheme which is theoretically limited in its order of accuracy [3].

The difficulty is to design an efficient polynomial reconstruction that would allow high-order accuracy while remaining cost-effective. A linear reconstruction operator applied on a single stencil in general will produce spurious oscillations in the vicinity of discontinuities. As multiphase flows systematically involve a severe discontinuity (e.g. density jumps of several orders of magnitude) it is crucial to mitigate this deficiency. For this purpose, we make use of weighted essentially non-oscillatory (WENO) schemes [49, 95, 106, 107, 131, 186, 251, 285, 286]. The WENO reconstruction is performed on unstructured meshes by applying a linear reconstruction

procedure to the various WENO stencils and combining the polynomials obtained for each stencil, according to the solution-dependent weights.

A feature of the methodology presented here is the tetrahedral decomposition of the mesh to handle the geometrical complexity of 3D mixed-element meshes. This makes our technique quite general and appropriate for most industrial problems.

6.3 Numerical formulation

6.3.1 Methodology for the linear reconstruction

The linear reconstruction is presented in this section for a scalar variable, however its extension to vector variables is straightforward. For each cell of the computational domain, the linear reconstruction procedure produces a polynomial representing the variable $u(x, y, z, t)$ everywhere in the cell. The stencil \mathcal{S} of cells E_j is used to generate the interpolating polynomial on the targeted cell E_0 . For convenience the first element of the stencil is the targeted cell. Hence:

$$\mathcal{S} = \bigcup_{j=0}^{jmax} E_j \quad (6.7)$$

For each cell of the mesh, the polynomial $p(x, y, z, t)$ is reconstructed from the cell-averages of the variables in the cells of the associated stencil \mathcal{S} . The reconstruction is performed with the constraint that the integral of the polynomial over the targeted cell equals the cell-average value in this cell. This conservation condition is expressed as:

$$\bar{u}_0 = \frac{1}{|E_0|} \iiint_{E_0} p(x, y, z) \, dx \, dy \, dz \quad (6.8)$$

The polynomial reconstruction in physical coordinates $\mathbf{x} = (x, y, z)$ on a general unstructured mesh requires the consideration of scaling effects. However, it is crucial for the generality of the method to undertake the reconstruction in a reference space $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ where scaling effects do not apply (see [49]). As well as elegantly

simplifying the polynomial reconstruction, working in a reference space prevents the cumbersome introduction of inaccurate scaling factors.

It is necessary to relate the physical coordinates to the reference coordinates by the mapping $\mathbf{x} = \mathbf{x}(\xi, \eta, \zeta)$. This operation requires the coordinates of four different vertices of the targeted element E_i : one for the origin of the frame, and three others — linked to the origin by an edge of E_i — to form the basis. For each cell in the mesh, the inverse of this cell-dependent mapping: $\boldsymbol{\xi} = \boldsymbol{\xi}(x, y, z)$ is applied to the targeted cell and its stencil. This leads to a stencil in the reference space:

$$\mathcal{S}' = \bigcup_{j=0}^{jmax} E'_j \quad (6.9)$$

Since the spatial average is not affected by the affine transformation $\boldsymbol{\xi} = \boldsymbol{\xi}(x, y, z)$, the conservation condition is also valid in the reference space. With $p(\xi, \eta, \zeta)$ defined as the outcome of the polynomial reconstruction in the reference space, we have:

$$\bar{u}_0 = \frac{1}{|E'_0|} \iiint_{E'_0} p(\xi, \eta, \zeta) \, d\xi \, d\eta \, d\zeta \quad (6.10)$$

To design a method for the determination of $p(\xi, \eta, \zeta)$, it is convenient to express the polynomial in a basis of polynomial functions: $\{\phi_k(\xi, \eta, \zeta)\}_{k=0\dots K}$. Introducing the degrees of freedom a_k , each associated to a given basis function, we write:

$$p(\xi, \eta, \zeta) = \bar{u}_0 + \sum_{k=1}^K a_k \phi_k(\xi, \eta, \zeta) \quad (6.11)$$

Therefore, the interpolating polynomial is completely defined by the set of $(K + 1)$ basis functions ϕ_k and their associated degrees of freedom a_k . Also, the integer K is related to the degree of the polynomial r by the expression:

$$K = \frac{(r + 1)(r + 2)(r + 3)}{6} - 1 \quad (6.12)$$

The basis functions must be constructed so that the conservation condition (6.10) is respected. This implies that the mean value of each basis function over E'_0 is null.

As in [254], we choose:

$$\phi_k = \psi_k - \frac{1}{|E'_0|} \iiint_{E'_0} \psi_k \, d\xi \, d\eta \, d\zeta \quad (6.13)$$

where $\{\psi_k\} = \xi, \eta, \zeta, \xi^2, \xi \cdot \eta, \dots, \zeta^r \quad k = 1, \dots, K$.

Knowing the basis functions, we can calculate the degrees of freedom a_k by requiring that the cell-averages of $p(\xi, \eta, \zeta)$ over each cell E'_j of the stencil \mathcal{S}' is equal to the corresponding cell-average of the variable: \bar{u}_j . This is expressed by the formula:

$$\bar{u}_j = \frac{1}{|E'_j|} \iiint_{E'_j} p(\xi, \eta, \zeta) \, d\xi \, d\eta \, d\zeta \quad j = 1, \dots, j_{max} \quad (6.14)$$

$$= \bar{u}_0 + \sum_{k=1}^K \left(\frac{1}{|E'_j|} \iiint_{E'_j} \phi_k(\xi, \eta, \zeta) \, d\xi \, d\eta \, d\zeta \right) a_k \quad j = 1, \dots, j_{max} \quad (6.15)$$

$$= \bar{u}_0 + \sum_{k=1}^K A_{jk} a_k \quad j = 1, \dots, j_{max} \quad (6.16)$$

where A_{jk} is the cell-average of the basis function ϕ_k over the cell E'_j of \mathcal{S}' .

Introducing $b_j = (\bar{u}_j - \bar{u}_0)$ in (6.16), the system of equations for the degrees of freedom a_k takes the matrix form:

$$\sum_{k=1}^K A_{jk} a_k = b_j \quad j = 1, \dots, j_{max} \quad (6.17)$$

It can be seen from (6.15) and (6.17) that the reconstruction matrix \mathcal{A} is solution independent. This means that the time-consuming operations required for the determination of the a_k can be pre-processed.

The solution of (6.17) provides the polynomial needed for the calculation of the inter-cell fluxes (see Section 6.2). The determination of the fluxes depends on the problem considered and on the Riemann solver chosen. The application of this numerical method to the transport of the level set is detailed in Section 6.4.

Generation of a central stencil

In [186, 187], the stencil generation was performed by constructing — for each cell of the domain — the most compact stencil \mathcal{S} in the physical coordinate system

$\mathbf{x} = (x, y, z)$. Then, all of the stencils \mathcal{S} were mapped to the reference coordinate system $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ to produce the stencils \mathcal{S}' . In addition, to identify the most compact stencil in the physical space we proposed a novel method that gathered layers of “point-neighbours” (see Figure 6.2) for the stencil selection.

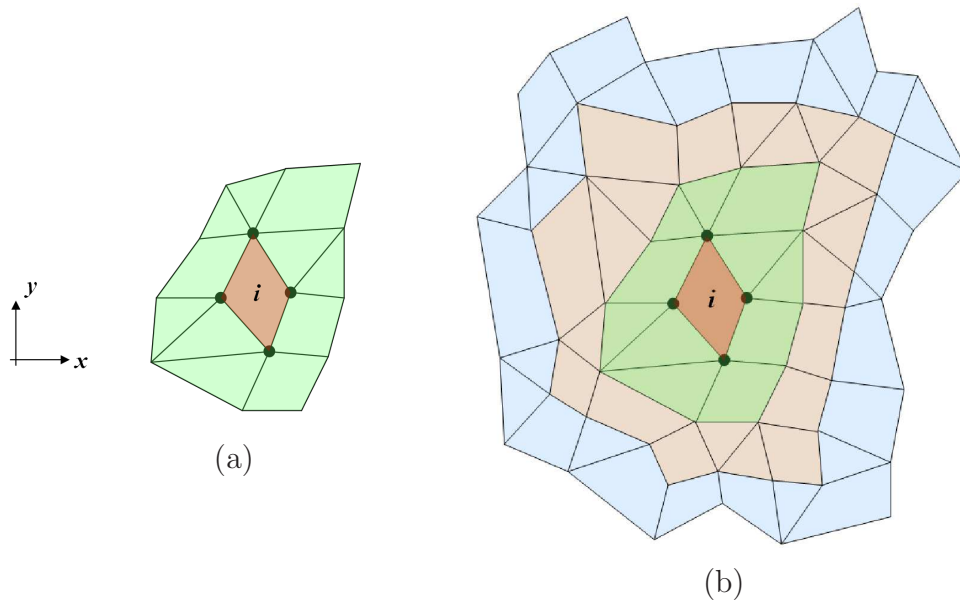


Figure 6.1: Point-neighbours approach in 2D — (a) Schematic of the point-neighbours (green) of a targeted cell (red); (b) Layers of cells added to the dynamic list at each iteration: in indigo, the cells added at the 1st iteration; in blue, the cells added at the 2nd iteration

Although this approach performed remarkably well on fully unstructured grids, numerical experiments showed that on anisotropic structured meshes — such as boundary layer meshes — this methodology lead to sub-optimal polynomial reconstruction of the solution. To mitigate this issue, the method has been modified, such that in the current algorithm:

- The selection of stencil cells is performed among layers of “face-neighbours”. Although this approach generally leads to less compact stencils, the set of cells produced extends further into the mesh for a given stencil size. As a result,

the selection algorithm has the option to reduce the redundancy of data on anisotropic structured meshes and the degrees of freedom are better defined.

- The compactness of the stencil is sought in the reference space ξ , such that the anisotropy of the mesh is cancelled out.

While still performing very well on fully unstructured grid, this novel approach systematically produces accurate polynomial reconstructions on anisotropic structured grid.

In the rest of Section 6.3, “face-neighbours” will be referred to as “neighbours” and the compactness of the stencil will be meant in the reference space. Besides, for the sake of simplicity, the pictures presented in this section have been generated assuming that the compactness in \mathbf{x} and ξ lead to the same stencils.

Number of cells of the stencil The purpose of the stencil is to provide a dataset (\bar{u}_j, E_j) for the polynomial reconstruction. Hence the number cells in the stencil j_{max} should be greater than or equal to the number of degrees of freedom K (see (6.12)). However, a stencil with the minimum number of cells may lead to unstable schemes on general meshes [49, 254]. Also, in the case $j_{max} = K$ the square matrix \mathcal{A} may not always be invertible for specific geometrical configurations. As a result, larger stencils are considered to maintain robustness. Typical sizes are: $1.5K$ in 2D and $2K$ in 3D [13, 110, 162].

Compact stencil in the mapped space In preparation for the determination of stencils for high-order schemes, we choose to pre-define the list of neighbours for each cell in the mesh. This list gathers all the cells that share a face with a given cell (see Figure 6.2 for an illustration in 2D). Creating such a data structure significantly simplifies the algorithm for stencil generation and increases the speed of the overall process.

Starting from the neighbours of the targeted cell E_0 , the method progresses by an

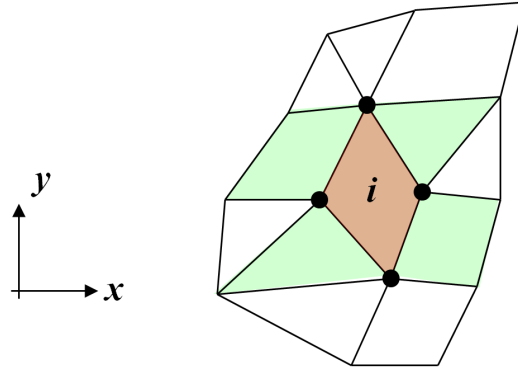


Figure 6.2: Schematic of the point-neighbours (green) of a targeted cell (red) — in 2D

iterative search over the neighbours of the cells in a dynamic list. The neighbours of all the cells in the list are consequently added to the list at each iteration. The cells added at a given iteration can be perceived as an additional layer of cells surrounding the cells already present in the list. This is illustrated in Figure 6.3. The loop stops when the total number of cells in the list is greater than or equal to j_{max} . Then, the cells in the list are sorted according to their centre-to-centre distance to the targeted cell E'_0 in ξ . In order to produce a compact stencil in the reference space, the j_{max} cells closest to E'_0 are selected for the stencil.

Mapping to a reference space Let us consider $M_0 = (x_0, y_0, z_0)$ to be any vertex of the targeted element E_0 and $M_1 = (x_1, y_1, z_1)$, $M_2 = (x_2, y_2, z_2)$, $M_3 = (x_3, y_3, z_3)$ to be three different vertices of E_0 linked to M_0 through an edge of E_0 such that the frame of reference $\mathcal{R}_0 = (M_0, \overrightarrow{M_0M_1}, \overrightarrow{M_0M_2}, \overrightarrow{M_0M_3})$ is direct. The mapping $\mathbf{x} = \mathbf{x}(\xi, \eta, \zeta)$ can be expressed as:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \mathcal{J} \begin{pmatrix} \xi \\ \eta \\ \zeta \end{pmatrix} \quad (6.18)$$

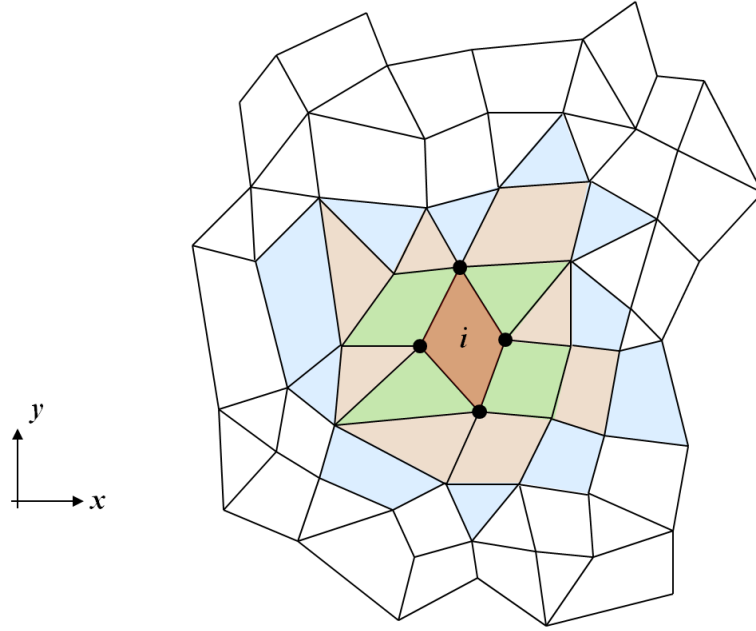


Figure 6.3: Layers of cells added to the dynamic list at each iteration: in indigo, the cells added at the 1st iteration; in blue, the cells added at the 2nd iteration — in 2D

where the Jacobian of the transformation \mathcal{J} is given by:

$$\mathcal{J} = \begin{pmatrix} x_1 - x_0 & x_2 - x_0 & x_3 - x_0 \\ y_1 - y_0 & y_2 - y_0 & y_3 - y_0 \\ z_1 - z_0 & z_2 - z_0 & z_3 - z_0 \end{pmatrix} \quad (6.19)$$

The algorithm applies the inverse of the mapping to the cell centres and vertices. For a generic point $P_{\mathbf{x}} = (x_P, y_P, z_P)$, the inverse transformation providing $P_{\boldsymbol{\xi}} = (\xi_P, \eta_P, \zeta_P)$ is:

$$\begin{pmatrix} \xi_P \\ \eta_P \\ \zeta_P \end{pmatrix} = \mathcal{J}^{-1} \begin{pmatrix} x_P - x_0 \\ y_P - y_0 \\ z_P - z_0 \end{pmatrix} \quad (6.20)$$

and the volumes of the transformed elements E'_j are given by:

$$|E'_j| = |\mathcal{J}^{-1}| |E_j| \quad (6.21)$$

Noting that the mapping gives $dx dy dz = |\mathcal{J}| d\xi d\eta d\zeta$ and using (6.21), we can prove that the conservation condition (6.10) is maintained after the affine transformation:

$$\begin{aligned} \iiint_{E_j} p(x, y, z) dx dy dz &= |E_j| \bar{u}_j \\ |\mathcal{J}| \iiint_{E'_j} p(\xi, \eta, \zeta) d\xi d\eta d\zeta &= |\mathcal{J}| |E'_j| \bar{u}_j \end{aligned} \quad (6.22)$$

In preparation of the mapping algorithm, we choose to pre-define the list of “edge-neighbours” for each point of each cell in the mesh. This list gathers all the points of a given cell that share an edge with a given point of this cell (see Figure 6.4). As with the list of “face-neighbours” for the stencil generation, this data structure significantly simplifies the implementation and speeds up the overall process.

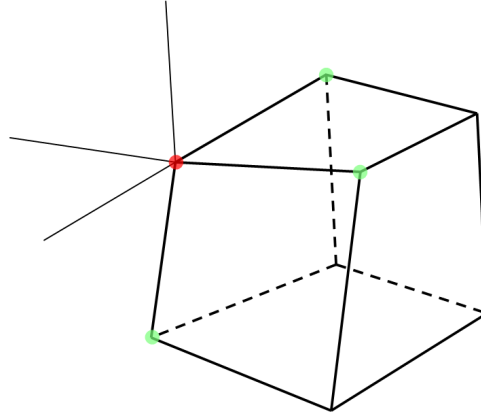


Figure 6.4: Schematic of the edge-neighbours (green vertices) of a given point (red vertex) in a given cell (thick lines) — for a hexahedral cell

Determination of the reconstruction matrix

As explained at the beginning of this section, the determination of the degrees of freedom a_k of the polynomial reconstruction involves the integration of the basis functions ϕ_k over the cells of the transformed stencil E'_j (see (6.15) and (6.16)).

Each of these integrations is an element of the reconstruction matrix \mathcal{A} involved in the equation for the degrees of freedom a_k (6.17). Combining (6.13) and (6.15), we get:

$$\begin{aligned} A_{jk} &= \frac{1}{|E'_j|} \iiint_{E'_j} \left(\psi_k - \frac{1}{|E'_0|} \iiint_{E'_0} \psi_k \, d\xi \, d\eta \, d\zeta \right) d\xi \, d\eta \, d\zeta \\ &= \frac{1}{|E'_j|} \iiint_{E'_j} \psi_k \, d\xi \, d\eta \, d\zeta - \frac{1}{|E'_0|} \iiint_{E'_0} \psi_k \, d\xi \, d\eta \, d\zeta \quad j = 1, \dots, j_{max} \end{aligned} \quad (6.23)$$

Therefore, equation (6.23) reduces the calculation of A_{jk} to a simple combination of monomial integrations over E'_j and E'_0 . According to Stroud [239], the most efficient way of calculating such multiple integrals over simple geometrical domains is to use Gauss-Legendre quadratures. However, Gaussian quadratures are only available over simple geometries such as unit n -simplexes or unit n -cubes. Even in the specific case of an unstructured hexahedral mesh, the trilinear mapping from a unit cube to a hexahedral cell is often non-invertible (see [262] for the conditions of non-degeneracy for hexahedral cells).

As a result, we choose to decompose each element of the mesh systematically into tetrahedra. Since a unit 3-simplex can always be mapped to a tetrahedron through an invertible transformation, the Gaussian quadratures can proceed. The method developed sums the Gaussian quadratures of the monomials calculated over all the tetrahedral sub-elements of a cell and then applies equation (6.23) to provide the matrix elements A_{jk} .

Tetrahedralisation of the mesh In order to ensure the convergence of the tetrahedralisation, we chose to split the polyhedral cells into tetrahedra that all have the centre of the original element as a vertex. In this manner, the polyhedron can be decomposed regardless of the way its faces are split and the tetrahedral decomposition is always possible. The faces of the elements (convex polygons) are split into as many triangles as they have sides using the face centre as common

vertex. This guarantee the convergence of the tetrahedral decomposition on generic meshes composed of convex polyhedra (see Figure 6.5).

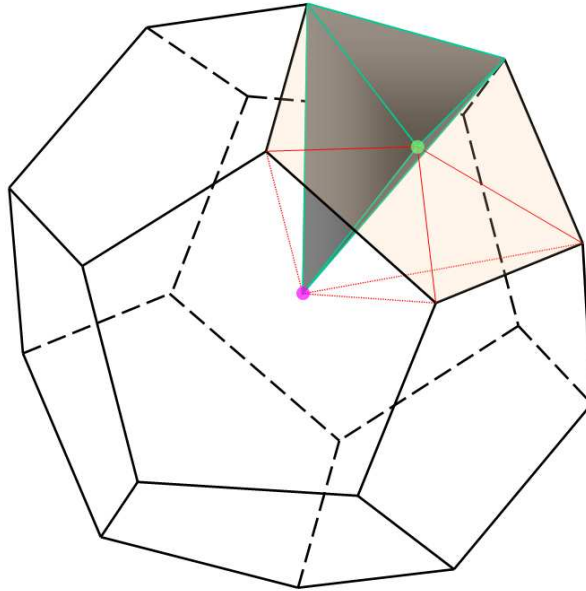


Figure 6.5: Tetrahedralisation of a convex polyhedron — In solid grey: a sub-element; in red: construction lines for the sub-elements; pink point: cell centre; shaded face: face decomposed; green point: centre of the face considered

To minimise the number of tetrahedra, the pyramids are still split into two sub-elements and the quadrilateral faces are split in two triangles. This method ensures the convergence of the tetrahedralisation as long as the quadrilateral faces are decomposed before the cells. This decomposition is illustrated for hexahedral and pentahedral cells in Figure 6.6.

Calculation of the degrees of freedom

Inversion of the reconstruction matrix As mentioned above in this section, in order to ensure the stability of the scheme on general meshes, the method collects more data than needed for the polynomial reconstruction by constructing a stencil of $j_{max} > K$ elements. This leads to the over-determination of the system of equation

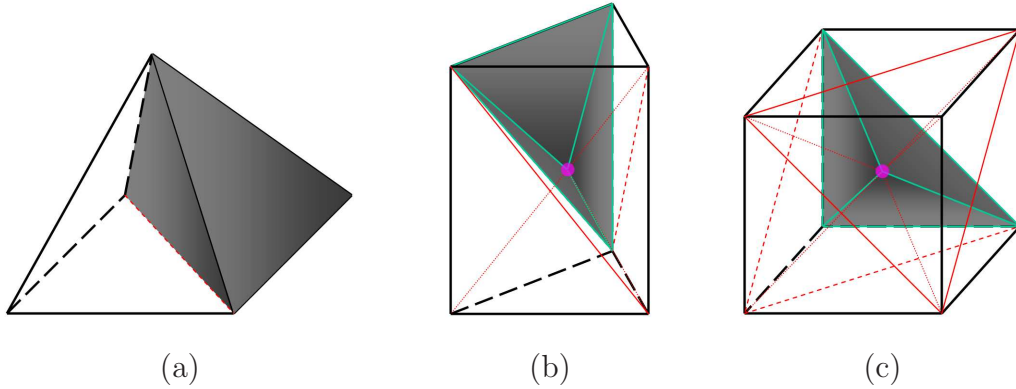


Figure 6.6: Tetrahedral decomposition ensuring convergence — In solid grey: a sub-element; in red: construction lines for the sub-elements; pink point: cell centre. (a) Pyramid into 2 tets; (b) Pentahedron into 8 tets; (c) Hexahedron into 12 tets

(6.17) for the degrees of freedom a_k .

Such a system is generally solved using a least-squares approach [49]. However, Titarev et al. [254] stated that such a method was not optimum for polynomial reconstruction of order higher than three. Indeed, they argue that the condition number of the linear system — produced by the application of the least-squares method for high-order polynomials — is approximately the square of the condition number of the matrix \mathcal{A} . This results in a potential lack of accuracy “for smooth problems and very fine meshes” [254].

Consequently, we follow the same approach as Titarev et al.: direct matrix factorisation of \mathcal{A} with a singular value decomposition (SVD) procedure. This is more demanding in computational time and storage than other methods, but it is considered as the most reliable method to compute the degrees of freedom, as it deals better with “errors in data, round-off errors and linear dependence” [57].

With \mathcal{A} a real $j_{max} \times K$ matrix with $j_{max} \geq K$, we can write [58]:

$$\mathcal{A} = \mathcal{U}\Sigma\mathcal{V}^T \quad (6.24)$$

where \mathcal{U} is the $j_{max} \times K$ matrix of the K orthonormalised eigenvectors associated

with the K largest eigenvalues of $\mathcal{A}\mathcal{A}^T$, such that $\mathcal{U}^T\mathcal{U} = \mathcal{I}_K$, \mathcal{V} is the $K \times K$ square matrix of orthonormalised eigenvectors of $\mathcal{A}^T\mathcal{A}$, such that $\mathcal{V}^T\mathcal{V} = \mathcal{I}_K$, and $\Sigma = \text{diag}(\sigma_1, \dots, \sigma_K)$ where σ_i are the non-negative square roots of the eigenvalues of $\mathcal{A}\mathcal{A}^T$. The matrices \mathcal{U} and \mathcal{V} can then be used to transform the equation (6.17) into an equivalent diagonal set of equations:

$$\begin{aligned}\mathcal{A}\mathbf{a} &= \mathbf{b} \\ (\mathcal{U}\Sigma\mathcal{V}^T)\mathbf{a} &= \mathbf{b} \\ \Sigma(\mathcal{V}^T\mathbf{a}) &= (\mathcal{U}^T\mathbf{b}) \\ \Sigma\bar{\mathbf{a}} &= \bar{\mathbf{b}}\end{aligned}\tag{6.25}$$

In order to illustrate the need to introduce a tolerance τ , below which singular values are neglected, let us assume that $\text{rank}(\mathcal{A}) = K$. Then, none of the singular values σ_k is equal to zero and one may solve equation (6.25) by setting:

$$\bar{a}_k = \frac{\bar{b}_k}{\sigma_k}\tag{6.26}$$

In the case of small σ_k , this may lead to undesirable sensitivity of the a_k to inaccuracies in the data and round-off errors. Hence, in order to compute a robust solution for the degrees of freedom, it is necessary to neglect all the singular values smaller than a given tolerance τ , representative of the accuracy of the data and the floating-point arithmetic. As a consequence, Σ is replaced by $\Sigma_\tau = \text{diag}(\sigma_{\tau,k})$ in (6.25) such that whenever $\sigma_k < \tau$, $\sigma_{\tau,k}$ is set to zero. Since it is always preferable to minimise the coefficients, whenever $\sigma_{\tau,k} = 0$, then $\bar{a}_k = 0$.

However, as the method calculates the degrees of freedom for all the cells of the mesh, storing the $(3 \times N_{cells})$ matrices produced by the SVD is memory consuming. To mitigate this, we chose to compute and store the ‘‘effective’’ Moore-Penrose pseudo-inverse \mathcal{A}_τ^\dagger of \mathcal{A} [57] which also further reduces the number of run-time

operations. This matrix can be defined in terms of the tolerance τ as:

$$\mathcal{A}_\tau^\dagger = \mathcal{V} \Sigma_\tau^\dagger \mathcal{U}^T$$

$$\text{where: } \Sigma_\tau^\dagger = \text{diag} \left(\sigma_{\tau,k}^\dagger \right) \quad \text{with: } \sigma_{\tau,k}^\dagger = \begin{cases} \frac{1}{\sigma_k} & \text{if } \sigma_k > \tau \\ 0 & \text{otherwise} \end{cases} \quad (6.27)$$

Run-time operations for the degrees of freedom All the above steps of the linear reconstruction are pre-computed and the results are stored. The run-time operations for the degrees of freedom then reduce to two trivial steps for every cell in the mesh:

1. The generation of the vector of data \mathbf{b} required for the calculation of the degrees of freedom in (6.17). The components of \mathbf{b} are computed from the cell-averages of the variable $\bar{u}_j(t)$ in the cells of the stencil \mathcal{S} at a given time t :

$$b_j = \bar{u}_j(t) - \bar{u}_0(t) \quad j = 1, \dots, j_{max} \quad (6.28)$$

2. The determination of the degrees of freedom from the “effective” pseudo-inverse \mathcal{A}_τ^\dagger and the vector of data \mathbf{b} :

$$\mathbf{a} = \mathcal{A}_\tau^\dagger \mathbf{b} \quad (6.29)$$

When $\text{rank}(\mathcal{A}) < K$ the equality in (6.17) no longer holds and $\mathcal{A}\mathbf{a}$ is only approximately equal to \mathbf{b} . There exists a set of solutions \mathfrak{S} which minimises the Euclidean norm of the residual:

$$\mathfrak{S} = \{ \mathbf{a} \mid |\mathcal{A}\mathbf{a} - \mathbf{b}|_2 = \min \} \quad (6.30)$$

It can be shown that the pseudo-inverse provides the shortest vector $\hat{\mathbf{a}}$ that minimises the norm [57, 174]:

$$\hat{\mathbf{a}} = \mathcal{A}_\tau^\dagger \mathbf{b} \quad \implies \quad \begin{cases} \hat{\mathbf{a}} \in \mathfrak{S} \\ \forall \mathbf{a} \in \mathfrak{S} : |\hat{\mathbf{a}}|_2 \leq |\mathbf{a}|_2 \end{cases} \quad (6.31)$$

Hence, the solution (6.29) is the least-squares solution to the system of equations (6.17).

6.3.2 Methodology for WENO schemes

As in Section 6.3.1, the WENO reconstruction procedure is presented in this section for a scalar variable, however its extension to vector variables is straightforward. The WENO reconstruction operator is based on several stencils \mathcal{S}_m : one central stencil \mathcal{S}_0 — generated in the same way as in Section 6.3.1 — and several sectoral stencils that cover all spatial directions in the vicinity of the targeted cell E_i . For convenience, the first element of the stencil is taken as the targeted cell. The appropriate minimal number of one-sided stencils that ensures the self-adaptation of the scheme near discontinuities is obtained by selecting one sectoral stencil per internal face of the cell.

For the cell faces near the boundaries of the domain, the associated one-sided stencil may have to be discarded as the sector may not encompass enough cells depending on the order of the scheme. As a result, the number of sectoral stencils N_{S_i} per mesh cell E_i varies according to the location of the cell with respect to the boundaries. The set of stencils for a given cell is then:

$$\mathcal{U} = \bigcup_{m=0}^{N_{S_i}} \mathcal{S}_m \quad (6.32)$$

Once the set of stencils is generated for all the cells of the mesh, the method proceeds with the linear polynomial reconstruction on each stencil \mathcal{S}_m of each cell E_i as described in Section 6.3.1. The WENO polynomial reconstruction is then given by the convex combination of all the polynomials $p_m(\xi, \eta, \zeta)$ associated with the stencils \mathcal{S}_m . Introducing the non-linear WENO weights w_m related to the stencils \mathcal{S}_m , we have:

$$p_{WENO}(\xi, \eta, \zeta) = \sum_{m=0}^{N_{S_i}} w_m p_m(\xi, \eta, \zeta) \quad (6.33)$$

where:

$$p_m(\xi, \eta, \zeta) = \bar{u}_0 + \sum_{k=1}^K a_k^{(m)} \phi_k(\xi, \eta, \zeta) \quad (6.34)$$

$$w_m = \frac{\gamma_m}{\sum_{m=0}^{N_{S_i}} \gamma_m} \quad \text{with:} \quad \gamma_m = \frac{d_m}{(\varepsilon + IS_m)^p} \quad (6.35)$$

The expression for the non-linear weights (6.35) involves the following parameters:

- d_m , the linear weight. Since the central stencil generally performs better for smooth solutions, following Dumbser and Käser [49] we choose to attribute a much larger linear weight to the central stencil.
- IS_m , the oscillation indicator which characterises the level of smoothness of the solution on the stencil \mathcal{S}_m . A smooth solution leads to a smaller oscillation indicator and therefore a larger weight.
- ε , a small positive number introduced in γ_m to prevent the denominator from becoming zero.
- p , the exponent of the oscillation indicator, devised to ensure that the contribution of non-smooth stencils vanishes.

We have chosen the following typical values for the WENO parameters [49, 110, 254]:

$$\begin{aligned} d_m &= \begin{cases} 10^3 & \text{if } m = 0 \\ 1 & \text{otherwise} \end{cases} \\ \varepsilon &= 10^{-6} \\ p &= 4 \end{aligned} \quad (6.36)$$

From the expressions for IS_m given in [49, 254], the following matrix formulation can be derived:

$$IS_m = \sum_{p=1}^K a_p^{(m)} \left(\sum_{q=1}^K B_{pq} a_q^{(m)} \right) \quad (6.37)$$

In (6.37), B_{pq} is an element of the oscillation indicator matrix \mathcal{B} . With r denoting the order of the linear polynomial reconstruction, and noting that $\gamma = \Lambda - \alpha - \beta$, the elements of \mathcal{B} are expressed as:

$$B_{pq} = \sum_{\Lambda=1}^r \sum_{\alpha=0}^{\Lambda} \sum_{\beta=0}^{\Lambda-\alpha} \iiint_{E'_0} \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \phi_p \times (\xi, \eta, \zeta) \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \phi_q (\xi, \eta, \zeta) \, d\xi \, d\eta \, d\zeta \quad (6.38)$$

As for the reconstruction matrix \mathcal{A} , it may be seen from (6.38) that the oscillation indicator matrix \mathcal{B} is solution independent, and hence the time consuming operations required for the determination of IS_m can be pre-processed. In addition, the elements B_{pq} involve the integration of polynomials on the targeted cell in the reference space E'_0 . As for the computation of \mathcal{A} we shall apply a Gauss-Legendre quadrature.

The WENO polynomial p_{WENO} can be expressed as a function of the modified degrees of freedom \tilde{a}_k by introducing (6.34) in equation (6.33) and using the condition $\sum_m w_m = 1$ [254]:

$$\begin{aligned} p_{WENO}(\xi, \eta, \zeta) &= \sum_{m=0}^{N_{S_i}} w_m \left(\bar{u}_0 + \sum_{k=1}^K a_k^{(m)} \phi_k(\xi, \eta, \zeta) \right) \\ &= \bar{u}_0 + \sum_{k=1}^K \left(\sum_{m=0}^{N_{S_i}} w_m a_k^{(m)} \right) \phi_k(\xi, \eta, \zeta) \\ &= \bar{u}_0 + \sum_{k=1}^K \tilde{a}_k \phi_k(\xi, \eta, \zeta) \end{aligned} \quad (6.39)$$

The combination of the $(N_{S_i} + 1)$ set of degrees of freedom $a_k^{(m)}$ into a single set of modified degrees of freedom \tilde{a}_k , as in [254], simplifies the algorithm and speeds up the computation of the WENO reconstruction.

Generation of the sectoral stencils

In order to ensure the self-adaptability of the scheme near discontinuities, additional one-sided stencils are associated with the targeted cell. To minimise the number of

stencils, Titarev et al. suggested the following guidelines [254]:

1. The stencils are disjoint (apart from the targeted cell).
2. The union of one-sided stencils covers the whole space surrounding the targeted cell.
3. The stencils are compact. We improved the methodology reported in [254] by choosing the stencils compact in the reference space ξ (see Section 6.3.1): i.e. in a given sector of the physical space the stencil gathers the j_{max} elements having the minimum centre-to-centre distance to the targeted cell in ξ .

As the mapping is trilinear, a given sector S_l of the physical space contains the elements E_j — images of the elements E'_j under the mapping $\mathbf{x} = \mathbf{x}(\xi, \eta, \zeta)$ — such that the inverse images E'_j are themselves contained in the sector of the reference space S'_l , inverse image of S_l under $\mathbf{x} = \mathbf{x}(\xi, \eta, \zeta)$. Therefore gathering elements in S_l for the stencil selection, is identical to gathering them in S'_l .

Definition of the sectors In order to remain as general as possible on a 3D mixed-element unstructured mesh, we aim to take as many compact stencils as there are internal faces FI_l of the considered element E_0 (i.e. faces of E_0 that are not on a domain boundary). For each face FI_l , we choose to define the sector — in which the stencil will be constructed — from the contour of FI_l and the centre C_0 of the cell E_0 . As illustrated in Figure 6.7, the cells selected to form the sector stencil have their centre in the portion of physical space:

- encompassing the centre of the face FI_l .
- delimited by the surface of the cone \mathcal{C}_l admitting C_0 as apex and the contour of FI_l as directrix.

This approach satisfies all of the three rules given above. In particular, the sectoral stencils cover all the spatial directions in the vicinity of the targeted cell. As a result, the “reverse sectors” of Käser et al. [49, 110] are unnecessary. This leads to a smaller number of stencils and thus to a faster method.

As mentioned at the beginning of Section 6.3.1, it is not always possible to find j_{max} cells in a sector when the internal face considered is close to a boundary (see Figure 6.7). The algorithm implemented systematically discards any stencil of less than j_{max} elements to ensure the robustness of the scheme in the vicinity of a boundary. It follows that the actual number of one-sided stencils of a given cell E_0 is at most the number of its internal faces N_{FI} :

$$N_{S_i} \leq N_{FI} \quad (6.40)$$

Mapping to the first octant In order to assess efficiently which cells have their centre lying in the sector, it is convenient to map the sector to the octant $(+, +, +)$ in a transformed space $\mathbf{X} = (X, Y, Z)$ so that only the cells whose centres have all positive coordinates in \mathbf{X} can be selected. However, such a mapping is only possible if the directrix of the cone \mathcal{C}_l encompassing the geometric sector is a triangle (see Figure 6.8). Taking the centre of the targeted cell $C_0 = (x_{C_0}, y_{C_0}, z_{C_0})$ and introducing \mathcal{J}_Q as the Jacobian of the affine transformation that maps the octant $(+, +, +)$ to the sector delimited by \mathcal{C}_l , we can write:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix} + \mathcal{J}_Q \begin{pmatrix} X \\ Y \\ Z \end{pmatrix} \quad (6.41)$$

where the matrix \mathcal{J}_Q is built from the coordinates of the three vertices of the triangle FI_l , the directrix of \mathcal{C}_l .

We extend this methodology to non-tetrahedral cells by splitting, into N_T triangles, all the internal faces FI_l of the mesh. The cells belonging to the sector S_l

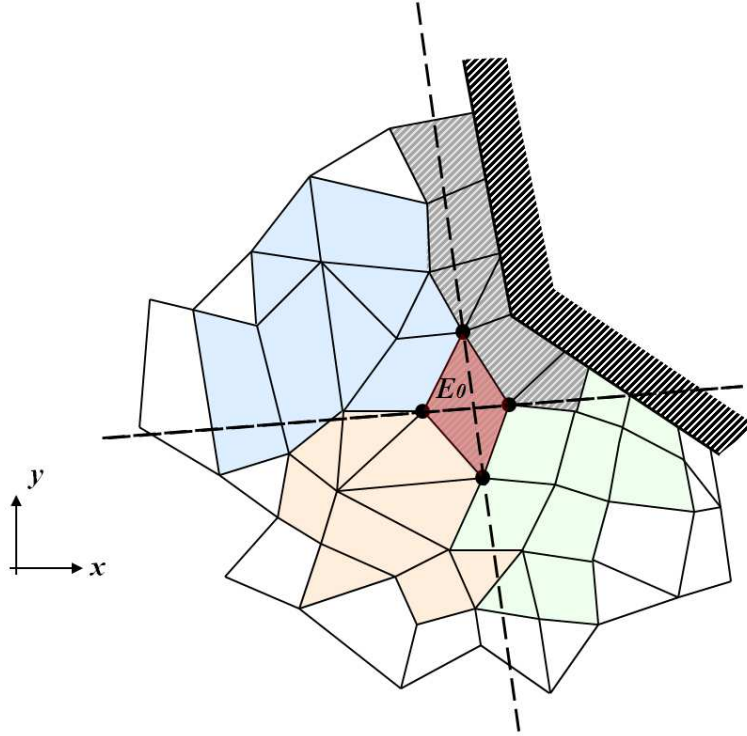


Figure 6.7: Ten-cells sectoral stencils of E_0 coloured by sector; discarded stencil hatched in grey; domain boundaries hatched in black — as produced by the fast search procedure in 2D

delimited by \mathcal{C}_i belong to the union of sub-sectors S_{l_i} delimited by the cones \mathcal{C}_{l_i} associated with the N_T triangles:

$$S_l = \bigcup_{i=1}^{N_T} S_{l_i} \quad (6.42)$$

The sub-sectors S_{l_i} are therefore successively mapped to the octant $(+, +, +)$ and for each potential stencil cell E_k , we produce and test a condition of inclusion of its centre C_k (expressed in $\mathbf{X}^{(i)}$: the transformed space associated to S_{l_i}) in the various sub-sectors. Using the symbol \sum to indicate the repeated use of the logical

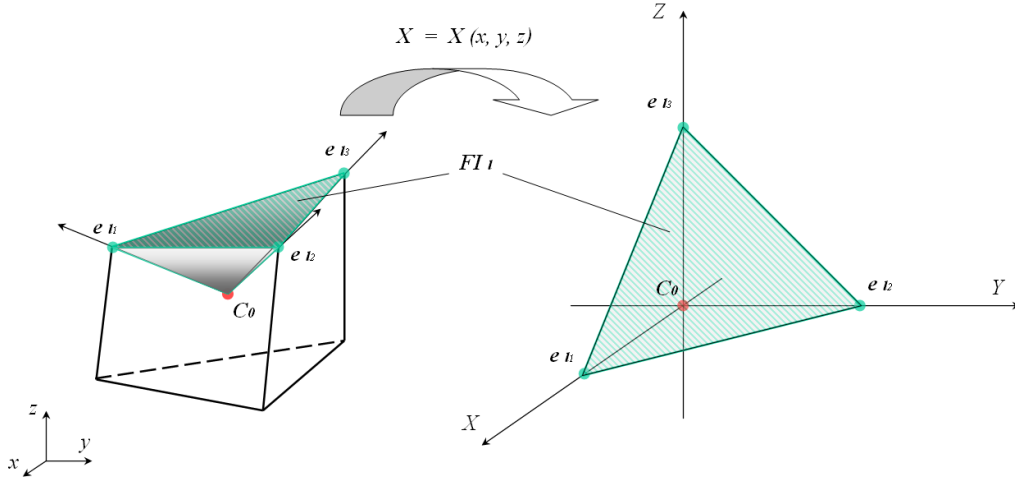


Figure 6.8: Mapping to the first octant (+, +, +)

disjunction, our condition reads:

$$Cond_k = \sum_{i=1}^{N_T} (C_k \in S_{l_i}) \quad (6.43)$$

$$= \sum_{i=1}^{N_T} \left(\text{pos} \left(X_{C_k}^{(i)} \right) \times \text{pos} \left(Y_{C_k}^{(i)} \right) \times \text{pos} \left(Z_{C_k}^{(i)} \right) \right) \quad (6.44)$$

where:

$$\forall x \in \mathbb{R} : \text{pos}(x) = \begin{cases} 1 & \text{if } x \geq 0 \\ 0 & \text{if } x < 0 \end{cases} \quad (6.45)$$

The above test guarantees that cells whose centres belong to several sub-sectors are added just once to the list of sector members. It avoids the need for a cumbersome and time-consuming iterative test against all the cells already identified as sector members. In order to optimise the procedure, the internal faces are split into a minimal number of triangles, such that $N_T = 1$ for triangular faces (no split), $N_T = 2$ for quadrilateral faces and $N_T = \textit{number of sides}$ for polygonal faces. As this split of the internal faces is performed in any case during the tetrahedralisation algorithm (see Section 6.3.1), the tetrahedral decomposition of the mesh is run prior to the generation of the one-sided stencils.

Selection of the stencil cells The cells whose centre belongs to a sector — the “sector members” — are added to the corresponding one-sided stencil if:

- They are not already part of another sectoral stencil of E_0 . This ensures that the one-sided stencils remain disjoint when the cell centre of a potential stencil element lies on the boundary of the sector. Such a particular case occurs frequently on Cartesian meshes.
- There are no more than $(j_{max} - 1)$ cells in this sector with a shorter centre-to-centre distance to E'_0 in ξ . This ensures the compactness of the stencil in the reference space.

The strategy adopted to produce a fast algorithm is to divide a very large central stencil in as many sectors as appropriate. Therefore, a much bigger temporary central stencil \mathcal{S}_U is produced (see Section 6.3.1), sorted according to the centre-to-centre distance to the targeted cell in ξ and finally split into the appropriate number of sectoral stencils (see Figure 6.9). Therefore, the search procedure starts with building a compact central stencil of N_U cells that should encompass a sufficient number of cells to produce all the one-sided stencils. N_U is defined by:

$$N_U = (N_{FI} + 1) j_{max} \tag{6.46}$$

The extra j_{max} cells resulting from the $(+1)$ in (6.46) provide a buffer of cells necessary to accommodate the selection of sectoral stencils near convoluted boundaries.

In order to remove the cumbersome and time-consuming test conditions related to the compactness and separation of the stencils, a dynamic list is initially identified to the stencil \mathcal{S}_U . Taking advantage of the fact that the cells in \mathcal{S}_U are already sorted according to their centre-to-centre distance to E'_0 in ξ , the membership of the cells in a given sector is tested sequentially for each cell in the list. This guarantees the compactness of the one-sided stencils in the reference space. The procedure stops either when j_{max} cells have been added to the stencil or when all the cells in the list

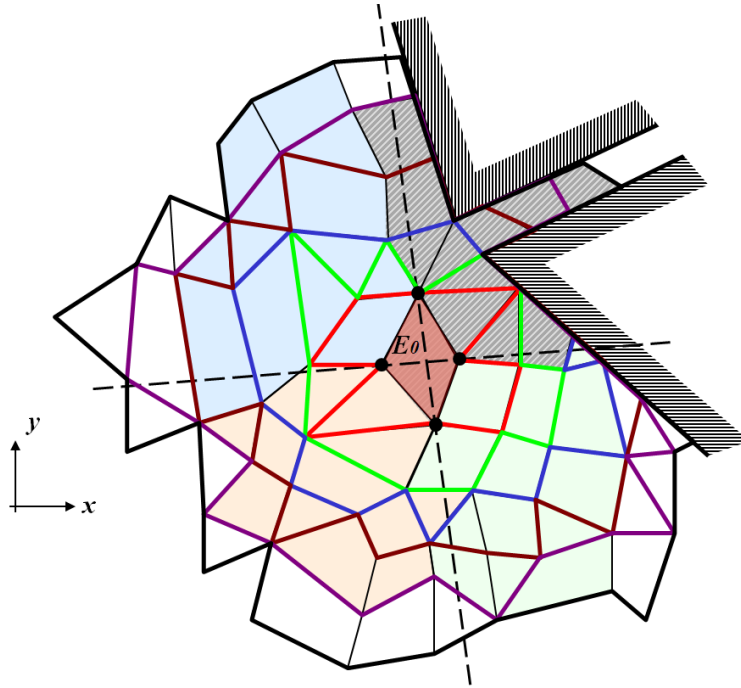


Figure 6.9: Thirteen-cells sectoral stencils of E_0 coloured by sector: the coloured thick lines indicate the progression of the search; discarded stencil hatched in grey; domain boundaries hatched in black — as produced by the fast search procedure in 2D

have been tested. The condition requiring disjoint stencils is ensured by removing the cells selected at each iteration. This fast search algorithm produces the best dataset for an accurate reconstruction of the solution, as the remaining stencils are very compact in ξ .

Determination of the oscillation indicator matrix

On a general unstructured mesh, the oscillation indicator matrix \mathcal{B} is solution independent and therefore can be pre-processed. We have derived a computationally

convenient expression for the elements of \mathcal{B} . Let us first rewrite equation (6.38) as:

$$B_{pq} = \sum_{\Lambda=1}^r \sum_{\alpha=0}^{\Lambda} \sum_{\beta=0}^{\Lambda-\alpha} \iiint_{E'_0} \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \phi_p(\xi, \eta, \zeta) \times \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \phi_q(\xi, \eta, \zeta) \, d\xi \, d\eta \, d\zeta \quad (6.38)$$

with: $\Lambda = \alpha + \beta + \gamma$

Simplified expression for B_{pq} The term B_{pq} is a triple sum of integrals over the targeted element E'_0 . Each integral is calculated through Gauss-Legendre quadratures over the N_T tetrahedral sub-elements T_l of E'_0 , resulting in the quadruple sum:

$$B_{pq} = \sum_{\Lambda=1}^r \sum_{\alpha=0}^{\Lambda} \sum_{\beta=0}^{\Lambda-\alpha} \sum_{l=1}^{N_T} \iiint_{T_l} \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \phi_p \times \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \phi_q \, d\xi \, d\eta \, d\zeta \quad (6.47)$$

The integrands involve the product of the partial derivatives of order (α, β, γ) , in (ξ, η, ζ) , taken on two different polynomial basis functions: ϕ_p and ϕ_q . As the basis functions are constructed to satisfy the conservation condition (6.12), the expression for the basis functions ϕ_k is simply the sum of a monomial ψ_k with a constant $C_{k,i}$ (dependent on the cell E'_i and the monomial ψ_k , see (6.15)):

$$\phi_k = \psi_k + C_{k,i} \quad (6.48)$$

Since the order of the partial derivative is greater than or equal to one, the constants disappear and we reduce each integrand $ID_{pq}^{\alpha,\beta,\gamma}$ to a product of two monomials:

$$B_{pq} = \sum_{\Lambda=1}^r \sum_{\alpha=0}^{\Lambda} \sum_{\beta=0}^{\Lambda-\alpha} \sum_{l=1}^{N_T} \iiint_{T_l} ID_{pq}^{\alpha,\beta,\gamma} \, d\xi \, d\eta \, d\zeta \quad (6.49)$$

$$= \sum_{\Lambda=1}^r \sum_{\alpha=0}^{\Lambda} \sum_{\beta=0}^{\Lambda-\alpha} \sum_{l=1}^{N_T} \iiint_{T_l} \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \psi_p \times \frac{\partial^{\Lambda}}{\partial \xi^{\alpha} \partial \eta^{\beta} \partial \zeta^{\gamma}} \psi_q \, d\xi \, d\eta \, d\zeta \quad (6.50)$$

Since a product of monomials is a monomial, the elements of \mathcal{B} are multiple sums of integrals of monomials over E'_0 . The integrations of the monomials of the highest degree D clearly occur for the basis functions of the highest degree when the order

of the partial derivative is lowest: $\Lambda_{min} = 1$. As a result, the highest degree D to be considered is expressed by:

$$\begin{aligned} D &= 2(r - \Lambda_{min}) \\ &= 2r - 2 \end{aligned} \quad (6.51)$$

The associated number of non-constant monomials of degree less than or equal to $(2r - 2)$ is then:

$$K_{IS} = \frac{r(4r^2 - 1)}{3} - 1 \quad (6.52)$$

Derivation of the integrands of B_{pq} Similarly to the determination of the elements of \mathcal{A} , the calculation of the B_{pq} involve a combination of integrals of monomials. However, in the computations of \mathcal{B} , the degree of the monomials to be integrated reaches $(2r - 2)$ and the integrations are only performed over the targeted cell E'_0 . In order to perform the integration, we express the integrand $ID_{pq}^{\alpha,\beta,\gamma}$ as a monomial. Introducing $(A, B, C) \in \llbracket 0, r \rrbracket^3$ such that $\{1 \leq A + B + C \leq r\}$, we can express the monomial $\psi_k(\xi, \eta, \zeta)$ by:

$$\psi_k(\xi, \eta, \zeta) = \xi^A \eta^B \zeta^C \quad (6.53)$$

Applying to ψ_k the partial derivative of order (α, β, γ) , in (ξ, η, ζ) , leads to:

$$\frac{\partial^\Lambda}{\partial \xi^\alpha \partial \eta^\beta \partial \zeta^\gamma} (\psi_k) = K_D \xi^{(A-\alpha)} \eta^{(B-\beta)} \zeta^{(C-\gamma)} \quad (6.54)$$

where:

$$K_D = \begin{cases} \frac{A!}{(A-\alpha)!} \times \frac{B!}{(B-\beta)!} \times \frac{C!}{(C-\gamma)!} & \text{if } (A \geq \alpha) \cap (B \geq \beta) \cap (C \geq \gamma) \\ 0 & \text{otherwise} \end{cases} \quad (6.55)$$

K_D can be expressed in a more convenient manner by using the function “pos” introduced in Section 6.3.1:

$$K_D = \text{pos}(A - \alpha) \times \text{pos}(B - \beta) \times \text{pos}(C - \gamma) \times \frac{A!}{(A-\alpha)!} \times \frac{B!}{(B-\beta)!} \times \frac{C!}{(C-\gamma)!} \quad (6.56)$$

Then, introducing: $(A_1, B_1, C_1, A_2, B_2, C_2) \in \llbracket 0, r \rrbracket^6$ such that $\{1 \leq A_i + B_i + C_i \leq r, i = 1, 2\}$ and making use of (6.56), we express the integrand $ID_{pq}^{\alpha, \beta, \gamma}$ as a monomial:

$$ID_{pq}^{\alpha, \beta, \gamma} = K_E \xi^{(A_1 + A_2 - 2\alpha)} \eta^{(B_1 + B_2 - 2\beta)} \zeta^{(C_1 + C_2 - 2\gamma)} \quad (6.57)$$

with:

$$K_E = \text{pos}(A_1 - \alpha) \times \text{pos}(B_1 - \beta) \times \text{pos}(C_1 - \gamma) \times \frac{A_1!}{(A_1 - \alpha)!} \times \frac{B_1!}{(B_1 - \beta)!} \times \frac{C_1!}{(C_1 - \gamma)!} \\ \times \text{pos}(A_2 - \alpha) \times \text{pos}(B_2 - \beta) \times \text{pos}(C_2 - \gamma) \times \frac{A_2!}{(A_2 - \alpha)!} \times \frac{B_2!}{(B_2 - \beta)!} \times \frac{C_2!}{(C_2 - \gamma)!} \quad (6.58)$$

Efficient computation of \mathcal{B} By inserting (6.57) in (6.49) we derive the final formulation for the elements B_{pq} :

$$B_{pq} = \sum_{\Lambda=1}^r \sum_{\alpha=0}^{\Lambda} \sum_{\beta=0}^{\Lambda-\alpha} \sum_{l=1}^{N_T} \iint\limits_{T_l} K_E \xi^{(A_1 + A_2 - 2\alpha)} \eta^{(B_1 + B_2 - 2\beta)} \zeta^{(C_1 + C_2 - 2\gamma)} d\xi d\eta d\zeta \quad (6.59)$$

It may be noted that:

- The K integrals of monomials of degrees up to r are readily available from the algorithm for computing the reconstruction matrix \mathcal{A} .
- The calculation of the different matrix elements B_{pq} generally involves many of the same monomial integrations over the targeted cell E'_0 .

Therefore we choose to store the list of K_{IS} integrals of monomials over E'_0 for each cell of the mesh. The elements of \mathcal{B} are then efficiently computed from this list of integrals by applying the formula (6.59). It is worth noting that only the integrals of monomials of degree higher than r have to be calculated, as the first K integrals of the list have already been computed (see Section 6.3.1).

Calculation of the modified degrees of freedom

Since the modified degrees of freedom \tilde{a}_k are a function of both the WENO weights w_m and the degrees of freedom $a_k^{(m)}$ associated to the stencil \mathcal{S}_m (see (6.39)), the

calculation of the \tilde{a}_k is performed at run time. This computation takes the following steps:

1. The generation of the vector of data \mathbf{b}_m for each stencil \mathcal{S}_m of each cell of the mesh. The components $b_j^{(m)}$ are computed from the cell averages of the solution $\bar{u}_j^{(m)}(t)$ in the cells of \mathcal{S}_m at a given time t :

$$b_j^{(m)} = \bar{u}_j^{(m)}(t) - \bar{u}_0^{(m)}(t) \quad j = 1, \dots, j_{max}; \quad m = 0, \dots, N_S \quad (6.60)$$

2. The determination of the $(N_S + 1)$ vectors of degrees of freedom \mathbf{a}_m from the effective pseudo-inverses $(\mathcal{A}_\tau^\dagger)^{(m)}$ of the reconstruction matrices and the vectors of data \mathbf{b}_m :

$$\mathbf{a}_m = (\mathcal{A}_\tau^\dagger)^{(m)} \mathbf{b}_m \quad m = 0, \dots, N_S \quad (6.61)$$

3. For each stencil \mathcal{S}_m , calculation of the smoothness indicator IS_m from the oscillation indicator matrix \mathcal{B} and the vector of degrees of freedom \mathbf{a}_m :

$$IS_m = \mathbf{a}_m^T (\mathcal{B} \mathbf{a}_m) \quad m = 0, \dots, N_S \quad (6.62)$$

4. Calculation of the WENO weights w_m from the smoothness indicators IS_m (see (6.37) and (6.38)).
5. For each cell of the mesh, calculation of the vector of modified degrees of freedom $\tilde{\mathbf{a}}$ from the $(N_S + 1)$ vector of degrees of freedom \mathbf{a}_m and the WENO weights w_m :

$$\tilde{\mathbf{a}} = \sum_{m=0}^{N_S} w_m \mathbf{a}_m \quad (6.63)$$

6.3.3 Determination of the numerical flux

With the details of the polynomial reconstruction procedure in mind, let us derive the flux calculation for the system of equations (6.1). This is a necessary step, since some of the simplifications relevant to tetrahedral grids [49] and 2D mixed-element

meshes [254] do not apply here. Recalling the equations derived in Section 6.2, we can express (6.1) in the following finite volume form:

$$\frac{d}{dt} \bar{\mathbf{U}}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} \mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+) d(F_l) = \mathbf{0} \quad (6.64)$$

Here $\mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+)$ represents the numerical flux in the direction normal to the face F_l (\mathbf{n}_l being the outward unit vector normal to F_l) as a function of the reconstructed solution on either side of F_l : \mathbf{U}^\pm . The superscripts “-” and “+” refer to the spatial limit respectively on the inside and the outside of the targeted cell E_i with respect to its face F_l . In particular, \mathbf{U}^- represents the reconstructed solution calculated on F_l using the polynomial interpolation of the solution in E_i and \mathbf{U}^+ represents the reconstructed solution calculated on F_l using the polynomial interpolation of the solution in the neighbouring cell E_{j_l} .

Hyperbolic systems of linear PDEs

The exact Riemann solver As in [49], we choose to use the exact Riemann solver to express the numerical flux between E_i and E_{j_l} . In order to express this flux, we first introduce the Jacobian matrix of the system in the normal direction \mathcal{J}_{N_l} . Noting that \mathcal{J}_X is the Jacobian of the vector of fluxes in the x direction $\mathbf{F}(\mathbf{U})$ we have (see [255]):

$$\begin{aligned} \frac{\partial}{\partial x} \mathbf{F}(\mathbf{U}) &= \frac{\partial \mathbf{F}}{\partial \mathbf{U}} \frac{\partial \mathbf{U}}{\partial x} \\ &= \mathcal{J}_X \frac{\partial \mathbf{U}}{\partial x} \end{aligned} \quad (6.65)$$

with \mathcal{J}_Y and \mathcal{J}_Z as the Jacobian matrices of the vectors of fluxes $\mathbf{G}(\mathbf{U})$ and $\mathbf{H}(\mathbf{U})$, we can re-write (6.1) as:

$$\frac{\partial}{\partial t} \mathbf{U} + \mathcal{J}_X \frac{\partial}{\partial x} \mathbf{U} + \mathcal{J}_Y \frac{\partial}{\partial y} \mathbf{U} + \mathcal{J}_Z \frac{\partial}{\partial z} \mathbf{U} = \mathbf{0} \quad (6.66)$$

Recalling $\mathbf{n}_l = (n_{lx}, n_{ly}, n_{lz})$ from Section 6.2, \mathcal{J}_{N_l} is expressed as:

$$\mathcal{J}_{N_l} = \mathcal{J}_X n_{lx} + \mathcal{J}_Y n_{ly} + \mathcal{J}_Z n_{lz} \quad (6.67)$$

In order to express the numerical flux \mathbf{A}_{n_l} , we introduce (after [255]) the diagonal matrix of eigenvalues $\Lambda_{N_l} = \text{diag}(\lambda_{l1}, \lambda_{l2}, \dots, \lambda_{lP})$ of \mathcal{J}_{N_l} and the left and right eigenvector matrices: \mathcal{L}_{N_l} and \mathcal{R}_{N_l} , so that $\mathcal{J}_{N_l} = \mathcal{L}_{N_l} \Lambda_{N_l} \mathcal{R}_{N_l}$. Defining the matrix absolute value operator: $|\mathcal{J}_{N_l}| = \mathcal{L}_{N_l} |\Lambda_{N_l}| \mathcal{R}_{N_l}$ with $|\Lambda_{N_l}| = \text{diag}(|\lambda_{l1}|, |\lambda_{l2}|, \dots, |\lambda_{lP}|)$, the numerical flux — associated with the exact Riemann solver — between E_i and E_j becomes [49, 255]:

$$\mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+) = \frac{1}{2} ((\mathcal{J}_{N_l} + |\mathcal{J}_{N_l}|) \mathbf{U}^- + (\mathcal{J}_{N_l} - |\mathcal{J}_{N_l}|) \mathbf{U}^+) \quad (6.68)$$

Computation of the numerical flux Introducing the expression (6.68) in the finite volume formulation (6.64) leads to:

$$\begin{aligned} \mathbf{0} = \frac{d}{dt} \bar{\mathbf{U}}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \frac{1}{2} (\mathcal{J}_{N_l} + |\mathcal{J}_{N_l}|) \iint_{F_l} \mathbf{U}^- d(F_l) \\ + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \frac{1}{2} (\mathcal{J}_{N_l} - |\mathcal{J}_{N_l}|) \iint_{F_l} \mathbf{U}^+ d(F_l) \end{aligned} \quad (6.69)$$

Since the reconstruction procedure relies on a mapping to avoid introducing cumbersome scaling factors, the reconstruction is performed on a reference space $\boldsymbol{\xi}$. As a result, the polynomial representing the solution \mathbf{U} is only known as a function of $\boldsymbol{\xi}$. Therefore, we relate the integrals in (6.69) to their counterparts over the faces F'_l of E'_i in the reference spaces $\boldsymbol{\xi}^-$ and $\boldsymbol{\xi}^+$ associated respectively to E'_i and E'_{j_l} :

$$\iint_{F_l} \mathbf{U}^\pm d(F_l) = \frac{|F_l|}{|F'_l|} \iint_{F'_l} \mathbf{U}^\pm(\boldsymbol{\xi}^\pm, t) d(F'_l) \quad (6.70)$$

As the transformation from the physical space to the reference space is affine, the ratio of square roots of the Gram determinants involved (see [117]) is constant and may be taken out of the integrals.

In the case of a vector variable \mathbf{U} , exactly the same basis functions ϕ_k are used for the polynomial reconstruction of all the components u_p of \mathbf{U} , since the functions ϕ_k are only dependent on the mesh. As each component of the vector variable can be

considered as a scalar variable, a set of k modified degrees of freedom \tilde{a}_k is calculated at each time step for every scalar variable u_p . Therefore there are as many modified degrees of freedom associated with ϕ_k as there are components in \mathbf{U} , and these are gathered in the vector $\tilde{\mathbf{a}}_k$.

Recalling the equation for the polynomial reconstruction of a scalar variable (6.39), the formulae for $\mathbf{U}^-(\boldsymbol{\xi}^-, t)$ and $\mathbf{U}^+(\boldsymbol{\xi}^+, t)$, at time $t = n\Delta t$, are:

$$\begin{cases} \mathbf{U}^-(\boldsymbol{\xi}^-, t) = (\overline{\mathbf{U}}_i)^{(n)} + \sum_{k=1}^K (\tilde{\mathbf{a}}_k)_i^{(n)} (\phi_k)_i(\boldsymbol{\xi}^-) \\ \mathbf{U}^+(\boldsymbol{\xi}^+, t) = (\overline{\mathbf{U}}_{j_i})^{(n)} + \sum_{k=1}^K (\tilde{\mathbf{a}}_k)_{j_i}^{(n)} (\phi_k)_{j_i}(\boldsymbol{\xi}^+) \end{cases} \quad (6.71)$$

Replacing $\mathbf{U}^-(\boldsymbol{\xi}^-, t)$ and $\mathbf{U}^+(\boldsymbol{\xi}^+, t)$ in (6.70) by their expressions in (6.71) leads to:

$$\begin{cases} \iint_{F_l} \mathbf{U}^- d(F_l) = |F_l| (\overline{\mathbf{U}}_i)^{(n)} + \frac{|F_l|}{|F'_l|} \sum_{k=1}^K \left((\tilde{\mathbf{a}}_k)_i^{(n)} \iint_{F'_l} (\phi_k)_i(\boldsymbol{\xi}^-) d(F'_l) \right) \\ \iint_{F_l} \mathbf{U}^+ d(F_l) = |F_l| (\overline{\mathbf{U}}_{j_i})^{(n)} + \frac{|F_l|}{|F'_l|} \sum_{k=1}^K \left((\tilde{\mathbf{a}}_k)_{j_i}^{(n)} \iint_{F'_l} (\phi_k)_{j_i}(\boldsymbol{\xi}^+) d(F'_l) \right) \end{cases} \quad (6.72)$$

In (6.72), the integrals of the basis functions $(\phi_k)_i(\boldsymbol{\xi}^-)$ and $(\phi_k)_{j_i}(\boldsymbol{\xi}^+)$ are independent of the solution \mathbf{U} . Hence the integrals of the k basis functions $(\phi_k)_i(\boldsymbol{\xi}^-)$ are pre-computed for all the faces F'_l of all the transformed elements E'_i . As a result, the integrals of $(\phi_k)_i(\boldsymbol{\xi}^-)$ and $(\phi_k)_{j_i}(\boldsymbol{\xi}^+)$ are readily available for the run-time calculation of the inter-cell flux between any neighbouring cells.

Surface integrals of the basis functions Recalling that the basis functions ϕ_k are constructed from the monomials ψ_k (see (6.13)), the expression for the k^{th} basis function of the element E_i : $(\phi_k)_i$ is:

$$(\phi_k)_i = \psi_k - \frac{1}{|E'_i|} \iiint_{E'_i} \psi_k d\xi d\eta d\zeta \quad (6.73)$$

Therefore, the integral of the basis function $(\phi_k)_i$ is expressed as:

$$\iint_{F'_l} (\phi_k)_i(\boldsymbol{\xi}^-) d(F'_l) = \iint_{F'_l} \psi_k(\boldsymbol{\xi}^-) d(F'_l) - \frac{|F'_l|}{|E'_i|} \iiint_{E'_i} \psi_k d\xi d\eta d\zeta \quad (6.74)$$

In (6.74), the triple integral of the monomials ψ_k over the volume E'_i is calculated during the linear reconstruction (see Section 6.3.1) and is therefore readily available. As in Section 6.3.1, in order to maximise the efficiency of the method, we choose to calculate the surface integral of the monomial ψ_k over the face F'_l with a Gaussian quadrature. Since the affine transformation of a convex polyhedral does not generally result in the standard domain required by Gauss-Legendre quadratures, we take advantage of the tetrahedralisation of the mesh and, for each face F'_l , sum the Gaussian quadratures produced for the N_{T_l} triangles T'_{lm} constituting F'_l :

$$\iint_{F'_l} \psi_k(\boldsymbol{\xi}^-) d(F'_l) = \sum_{m=1}^{N_{T_l}} \iint_{T'_{lm}} \psi_k(\boldsymbol{\xi}^-) d(T'_{lm}) \quad (6.75)$$

Hyperbolic systems of non-linear PDEs

In the general case, the flux $\mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+)$ (see (6.64)) varies along the face F_l . Consequently, as we perform the integration of the flux with a Gauss-Legendre quadrature, $\mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+)$ is evaluated at each Gauss point \mathbf{x}_β of each triangle T_{lm} constituting the face F_l .

The solution being reconstructed in the mapped space, the integral of the flux are calculated over the transformed face F'_l . Re-writing (6.64) with the integral expressed over F'_l leads to:

$$\frac{d}{dt} \bar{\mathbf{U}}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \frac{|F_l|}{|F'_l|} \iint_{F'_l} \mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+) d(F'_l) = \mathbf{0} \quad (6.76)$$

For each point of the Gaussian quadrature, the reconstructed solutions on either side of the face, \mathbf{U}_β^- and \mathbf{U}_β^+ , are evaluated at the mapped Gauss point in their respective transformed space: $\boldsymbol{\xi}_\beta^-$ and $\boldsymbol{\xi}_\beta^+$. Introducing the weights of the Gaussian

quadrature k_β and the Riemann solver $\widehat{\mathbf{A}}$, the integral of the flux can be expressed as follows:

$$\begin{aligned} \iint_{F'_l} \mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+) d(F'_l) &= \sum_{m=1}^{N_{T_l}} \iint_{T'_{lm}} \mathbf{A}_{n_l}(\mathbf{U}^-, \mathbf{U}^+) d(F'_l) \\ &= \sum_{m=1}^{N_{T_l}} \sum_{\beta=1}^{N_\beta} k_\beta \widehat{\mathbf{A}}(\mathbf{U}_\beta^-, \mathbf{U}_\beta^+) \end{aligned} \quad (6.77)$$

As suggested by the equations above, the integrals of the basis functions can no longer be pre-processed when solving non-linear PDEs. However, the values of the basis functions on each Gauss point can be pre-computed to save computational run time.

6.4 Application to the level set equation

6.4.1 Finite volume formulation of the level set equation

The WENO scheme presented in the previous section is to be used in the transport of the level set scalar function φ for application to multiphase flows. Introducing the velocity $\mathbf{u} = (u, v, w)$, the transport equation for φ may be expressed as:

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \quad (6.78)$$

Assuming an incompressible flow we have $\nabla \cdot \mathbf{u} = 0$, so that (6.78) can be re-written as a hyperbolic conservation law:

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{u}) = 0 \quad (6.79)$$

In terms of the coordinates (x, y, z) , (6.79) becomes

$$\frac{\partial \varphi}{\partial t} + \frac{\partial}{\partial x} (u\varphi) + \frac{\partial}{\partial y} (v\varphi) + \frac{\partial}{\partial z} (w\varphi) = 0 \quad (6.80)$$

Comparing equation (6.80) to equation (6.1) provides the equalities:

$$\left\{ \begin{array}{l} F(\varphi) = u\varphi \\ G(\varphi) = v\varphi \\ H(\varphi) = w\varphi \end{array} \right. \implies \left\{ \begin{array}{l} \mathbf{A} = (F, G, H) \\ = (u\varphi, v\varphi, w\varphi) \\ A_{n_l}(\varphi^-, \varphi^+) = \mathbf{A} \cdot \mathbf{n}_l \end{array} \right. \quad (6.81)$$

In (6.81), φ^- represents the reconstructed level set function calculated on F_l using polynomial interpolation of the solution in E_i , while φ^+ represents the reconstructed level set function calculated on F_l using polynomial interpolation of the solution in the neighbouring cell E_{j_l} .

From (6.81) and (6.64), we express the transport equation for the level set in the finite volume form:

$$\frac{d}{dt}\bar{\varphi}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} A_{n_l}(\varphi^-, \varphi^+) d(F_l) = 0 \quad (6.82)$$

The normal component of the flux A_{n_l} for the level set equation can be obtained by expressing (6.78) in terms of the coordinates to give:

$$\frac{\partial \varphi}{\partial t} + u \frac{\partial \varphi}{\partial x} + v \frac{\partial \varphi}{\partial y} + w \frac{\partial \varphi}{\partial z} = 0 \quad (6.83)$$

Identifying equation (6.83) with equation (6.66) provides the equalities:

$$\left\{ \begin{array}{l} J_X = u \\ J_Y = v \\ J_Z = w \end{array} \right. \implies \left\{ \begin{array}{l} J_{N_l} = un_{lx} + vn_{ly} + wn_{lz} \\ = \mathbf{u} \cdot \mathbf{n}_l \\ = u_{n_l} \end{array} \right. \quad (6.84)$$

Using (6.84) in (6.68), the normal component of the flux may be expressed as:

$$A_{n_l}(\varphi^-, \varphi^+) = \frac{1}{2} ((u_{n_l} + |u_{n_l}|) \varphi^- + (u_{n_l} - |u_{n_l}|) \varphi^+) \quad (6.85)$$

6.4.2 The Riemann problem for the level set equation

Simple manipulations of (6.83) demonstrates its rotational invariance (see Chapter 3 of [255]) according to:

$$\mathbf{A} \cdot \mathbf{n}_l = (F, G, H) \cdot \mathbf{n}_l = n_{lx}F + n_{ly}G + n_{lz}H = \widehat{F} \quad (6.86)$$

where \widehat{F} is the flux vector expressed in the direction n_l , the first axis of the rotated Cartesian frame (n_l, s_l, t_l) . The expression for \widehat{F} is:

$$\widehat{F} = (\mathbf{u} \cdot \mathbf{n}_l) \varphi = u_{n_l} \varphi \quad (6.87)$$

In our three-dimensional space, the rotated Cartesian frame is obtained by applying successively two rotations around the angles $\theta_{l1} \in [0, 2\pi]$ and $\theta_{l2} \in [0, \pi]$. The equation (6.86) is valid $\forall (\theta_{l1}, \theta_{l2}) \in [0, 2\pi] \times [0, \pi]$ (see Figure 6.10 for an illustration in 2D). Since we are dealing with a single scalar equation, the rotation matrix $\mathcal{T}_l(\theta_{l1}, \theta_{l2})$ (see [20, 255]) and its inverse both reduce to the scalar unity. From (6.81) and (6.82), the finite volume formulation of (6.83) can be expressed as:

$$\frac{d}{dt} \varphi_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} (F, G, H) \cdot \mathbf{n}_l \, d(F_l) = 0 \quad (6.88)$$

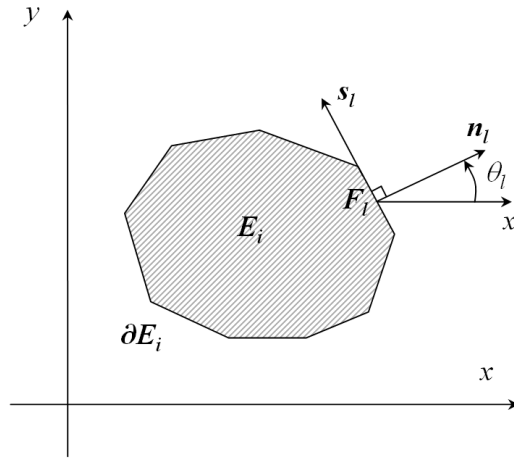


Figure 6.10: Rotated Cartesian frame in 2D: (n_l, s_l) — the first axis n_l is orthogonal to the arbitrary boundary ∂E_i of the control volume E_i

Using the rotational invariance of (6.83) we re-write (6.88) in the rotated Cartesian frame (n_l, s_l, t_l) , where the direction n_l is normal to the inter-cell boundary F_l . Noting \widehat{F} , \widehat{G} and \widehat{H} , the flux vectors respectively in the directions n_l , s_l and t_l , we

have:

$$\begin{aligned}
 \frac{d}{dt}\bar{\varphi}_i &= -\frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} (\widehat{F}, \widehat{G}, \widehat{H}) \cdot \mathbf{n}_l \, d(F_l) \\
 &= -\frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} (\widehat{F}, \widehat{G}, \widehat{H}) \cdot (1, 0, 0)^T \, d(F_l) \\
 &= -\frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} \widehat{F} \, d(F_l)
 \end{aligned} \tag{6.89}$$

Therefore, as mentioned in [255], the numerical fluxes across the L_i faces F_l of the element E_i result from the equation written in the rotated frame (n_l, s_l, t_l) . Consequently, the flux across F_l is given by the one-dimensional equation:

$$\frac{\partial \varphi}{\partial t} + \frac{\partial \widehat{F}}{\partial n_l} = 0 \tag{6.90}$$

Equation (6.90) leads to the Riemann problem:

$$\left. \begin{array}{l}
 \text{PDE:} \quad \frac{\partial \varphi}{\partial t} + u_{n_l} \frac{\partial \varphi}{\partial n_l} = 0 \\
 \text{IC:} \quad \varphi(n_l, 0) = \varphi_0(n_l) = \begin{cases} \varphi^- & \text{if } n_l < 0 \\ \varphi^+ & \text{if } n_l > 0 \end{cases}
 \end{array} \right\} \tag{6.91}$$

Equation (6.91) admits an exact solution:

$$\varphi(n_l, t) = \varphi_0(n_l - u_{n_l}t) = \begin{cases} \varphi^- & \text{if } n_l - u_{n_l}t < 0 \\ \varphi^+ & \text{if } n_l - u_{n_l}t > 0 \end{cases} \tag{6.92}$$

so that the flux A_{n_l} (see (6.81)) across F_l (i.e. at $n_l = 0$ with $t > 0$), reads:

$$A_{n_l}(\varphi^-, \varphi^+) = \begin{cases} u_{n_l} \varphi^- & \text{if } u_{n_l} > 0 \\ u_{n_l} \varphi^+ & \text{if } u_{n_l} < 0 \end{cases} \tag{6.93}$$

The formulation of the flux in (6.93) is equivalent to the expression for $A_{n_l}(\varphi^-, \varphi^+)$ given in (6.85). Thus, the finite volume formulation of the level set equation on 3D unstructured grids has been related to its associated Riemann problem.

6.5 Application to the Burgers' equation

6.5.1 Finite volume formulation of the Burgers' equation

In this section we extend the application of the WENO scheme to the solution of the Burgers' equation. Considering the variable φ and introducing the vector $\mathbf{v} = (a, b, c)$ fixed in \mathbb{R}^d , the Burgers' equation for φ may be expressed as:

$$\frac{\partial \varphi}{\partial t} + \mathbf{v} \cdot \nabla \left(\frac{\varphi^2}{2} \right) = 0 \quad (6.94)$$

As \mathbf{v} is fixed in \mathbb{R}^d , (6.94) can be re-written as a hyperbolic conservation law:

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot \left(\frac{\varphi^2}{2} \mathbf{v} \right) = 0 \quad (6.95)$$

In terms of the coordinates (x, y, z) , (6.95) becomes

$$\frac{\partial \varphi}{\partial t} + \frac{\partial}{\partial x} \left(a \frac{\varphi^2}{2} \right) + \frac{\partial}{\partial y} \left(b \frac{\varphi^2}{2} \right) + \frac{\partial}{\partial z} \left(c \frac{\varphi^2}{2} \right) = 0 \quad (6.96)$$

Comparing equation (6.96) to equation (6.1) provides the equalities:

$$\left\{ \begin{array}{l} F(\varphi) = a \frac{\varphi^2}{2} \\ G(\varphi) = b \frac{\varphi^2}{2} \\ H(\varphi) = c \frac{\varphi^2}{2} \end{array} \right. \implies \left\{ \begin{array}{l} \mathbf{A} = (F, G, H) \\ \phantom{\mathbf{A}} = \left(a \frac{\varphi^2}{2}, b \frac{\varphi^2}{2}, c \frac{\varphi^2}{2} \right) \\ A_{n_l}(\varphi^-, \varphi^+) = \mathbf{A} \cdot \mathbf{n}_l \end{array} \right. \quad (6.97)$$

The Burgers' equation can then be expressed in the following finite volume form:

$$\frac{d}{dt} \bar{\varphi}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} A_{n_l}(\varphi^-, \varphi^+) d(F_l) = 0 \quad (6.98)$$

6.5.2 The Riemann problem for the Burgers' equation

As in Section 6.4, simple manipulations of (6.96) demonstrates its rotational invariance according to:

$$\mathbf{A} \cdot \mathbf{n}_l = (F, G, H) \cdot \mathbf{n}_l = n_{lx}F + n_{ly}G + n_{lz}H = \widehat{F} \quad (6.99)$$

where \widehat{F} is the flux vector expressed in the direction n_l , the first axis of the rotated Cartesian frame (n_l, s_l, t_l) . The expression for \widehat{F} is:

$$\widehat{F} = (\mathbf{v} \cdot \mathbf{n}_l) \frac{\varphi^2}{2} = v_{n_l} \frac{\varphi^2}{2} \quad (6.100)$$

Therefore, the flux across F_l is given by the one-dimensional equation:

$$\frac{\partial \varphi}{\partial t} + \frac{\partial \widehat{F}}{\partial n_l} = 0 \quad (6.101)$$

Equation (6.101) leads to the Riemann problem:

$$\left. \begin{array}{l} \text{PDE:} \quad \frac{\partial \varphi}{\partial t} + \frac{\partial}{\partial n_l} \left(v_{n_l} \frac{\varphi^2}{2} \right) = 0 \\ \text{IC:} \quad \varphi(n_l, 0) = \varphi_0(n_l) = \begin{cases} \varphi^- & \text{if } n_l < 0 \\ \varphi^+ & \text{if } n_l > 0 \end{cases} \end{array} \right\} \quad (6.102)$$

For the PDE (6.102), the characteristic speed $\lambda(\varphi)$ is given by:

$$\lambda(\varphi) = \frac{d\widehat{F}}{d\varphi} = v_{n_l} \varphi \quad (6.103)$$

Equation (6.102) admits an exact solution:

$$\left. \begin{array}{l} \text{If } \lambda(\varphi^-) > \lambda(\varphi^+) : \quad \varphi(n_l, t) = \begin{cases} \varphi^- & \text{if } n_l - St < 0 \\ \varphi^+ & \text{if } n_l - St > 0 \end{cases} \\ \text{with: } S = \frac{\Delta \widehat{F}}{\Delta \varphi} = \frac{v_{n_l}}{2} (\varphi^- + \varphi^+) \\ \text{If } \lambda(\varphi^-) \leq \lambda(\varphi^+) : \quad \begin{cases} \varphi(n_l, t) = \varphi^- & \text{if } \frac{n_l}{t} \leq \lambda(\varphi^-) \\ \lambda(\varphi) = \frac{n_l}{t} & \text{if } \lambda(\varphi^-) < \frac{n_l}{t} < \lambda(\varphi^+) \\ \varphi(n_l, t) = \varphi^+ & \text{if } \frac{n_l}{t} \geq \lambda(\varphi^+) \end{cases} \end{array} \right\} \quad (6.104)$$

so that the flux A_{n_l} (see (6.97)) across F_l (i.e. at $n_l = 0$ with $t > 0$), reads:

$$\left. \begin{aligned}
 &\text{If } \lambda(\varphi^-) > \lambda(\varphi^+) : & A_{n_l}(\varphi^-, \varphi^+) &= \begin{cases} v_{n_l} \frac{(\varphi^-)^2}{2} & \text{if } S > 0 \\ v_{n_l} \frac{(\varphi^+)^2}{2} & \text{if } S < 0 \end{cases} \\
 &\text{with: } S = \frac{v_{n_l}}{2}(\varphi^- + \varphi^+) \\
 &\text{If } \lambda(\varphi^-) \leq \lambda(\varphi^+) : & A_{n_l}(\varphi^-, \varphi^+) &= \begin{cases} v_{n_l} \frac{(\varphi^-)^2}{2} & \text{if } 0 \leq v_{n_l} \varphi^- \\ 0 & \text{if } v_{n_l} \varphi^- < 0 < v_{n_l} \varphi^+ \\ v_{n_l} \frac{(\varphi^+)^2}{2} & \text{if } 0 \geq v_{n_l} \varphi^+ \end{cases}
 \end{aligned} \right\} \quad (6.105)$$

6.6 Performance of the scheme

The approach described above was implemented in parallel and in C++ using the framework provided by the open source CFD toolkit OpenFOAM. In order to reach the appropriate level of accuracy in parallel, the size of the halo surrounding each processor's sub-domain varies with the order of the polynomial reconstruction.

In all the simulations presented, the time discretisation is performed with a third-order Runge-Kutta scheme [231]. As we test WENO schemes of different order, let us introduce WENO r , the WENO scheme based on a polynomial reconstruction of order r .

6.6.1 Level set test cases

In all of the test cases considered in this section, we are advecting the level set scalar function φ according to equation (6.78) only. No re-distancing is applied as we are concerned mainly with assessing the performance of the WENO scheme.

Two-dimensional test cases

Two-dimensional meshes For these simulations, the mesh is made up of a single layer of three-dimensional elements: wedges for the “triangular” mesh, hexahedra for the Cartesian mesh and both wedges and hexahedra for the hybrid mesh. As a result, the mesh is in fact three-dimensional as the code employed manages only 3D elements.

A front view of the three types of mesh used is given in Figure 6.11 for the resolution $L/64$, with L being the length of the domain. Each two-dimensional mesh is shown on the left with a section of the corresponding three-dimensional mesh on the right. Three different resolutions have been considered: $L/64$, $L/128$ and $L/256$. The unstructured meshes have been generated so that — for a given resolution — the sides of the elements have the same length. As a result, the mesh density is higher for the “triangular” mesh and in the triangular region of the hybrid mesh.

Translation of a slotted disk The simulation settings for the solid body rotation of a slotted disk are given in Section 4.2.1 and the results for this problem are presented in Figure 6.12. This figure provides the solution obtained with the WENO3 scheme after one revolution for three different resolutions of $L/64$ (red line), $L/128$ (green line) and $L/256$ (blue line) on three types of grids: Cartesian, triangular and hybrid (top to bottom). The results illustrate the convergence of the solution under grid refinement regardless of the type of mesh. Indeed, for all test cases, the level set contour for the resolution $L/256$ is closer to the initial contour in both the smooth circular part of the disk and in the region of the slot where both the width of the slot and the sharp corners are well captured.

The results are comparable across mesh types and, as intended, no significant degradation was observed when running the test case on general unstructured grids. On the triangular grid, the accuracy of the scheme is slightly better as this mesh is

characterised by a higher density of elements for the same edge length.

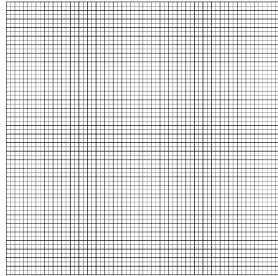
Disk in a deformation field The simulation settings for the disk in the deformation field are given in Section 4.2.1 and the results for this problem are presented in Figure 6.12. This figure provides the solution obtained with the WENO3 scheme at $t = 3s$ for three different resolutions of $L/64$ (red line), $L/128$ (green line) and $L/256$ (blue line) on three types of grids: Cartesian, triangular and hybrid (top to bottom). The solution obtained with the WENO3 scheme on the triangular grid (highest mesh density) of resolution $L/256$ has been taken as the reference case (black line), as it provides the results closest to the “exact solution” as obtained with the marker particle method by Rider and Kothe [201] or with the hybrid particle level set method by Enright et al. [53].

For all test cases, under grid refinement the scheme demonstrates a greater ability to capture thin ligaments, regardless of the type of mesh. Indeed, when the mesh resolution is increased, the tail of the spiral becomes systematically longer. As with the previous test case, no significant degradation of the results were observed on general unstructured grids. It is worth also noting that the results obtained with this WENO3 scheme without re-distancing compare very well with existing methods that use WENO schemes of order five together with re-distancing (see [37, 246]).

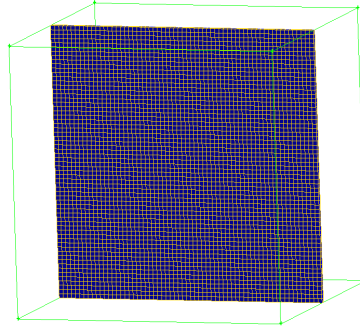
NB: In order to compare the performance of the scheme with the work of previous authors, the above two-dimensional test cases have been run with typical settings. However, the comparison of the zero level set computed at a later time should further differentiate the high-order schemes from the low-order ones [3].

Cartesian meshes

4096 cells (hexes)

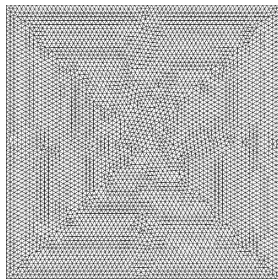


262144 cells (hexes)

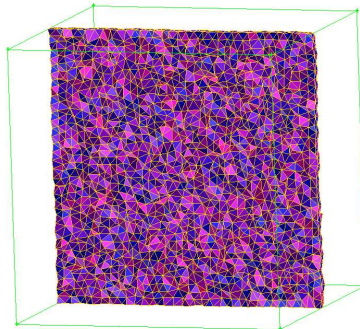


Triangular (left) and tetrahedral (right) meshes

9192 cells (wedges)

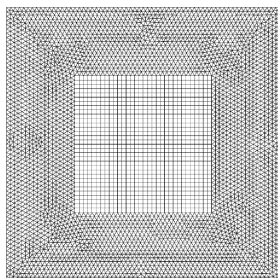


250704 cells (tets)



Hybrid meshes

7976 cells (hexes & wedges)



234843 cells (hexes, pyramids & tets)

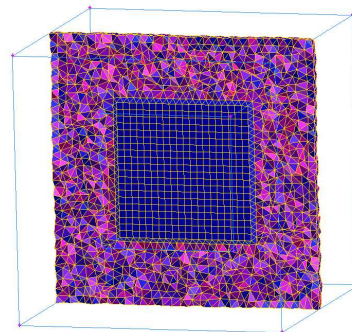


Figure 6.11: Meshes for the level set 2D (left column) and 3D (right column) test cases

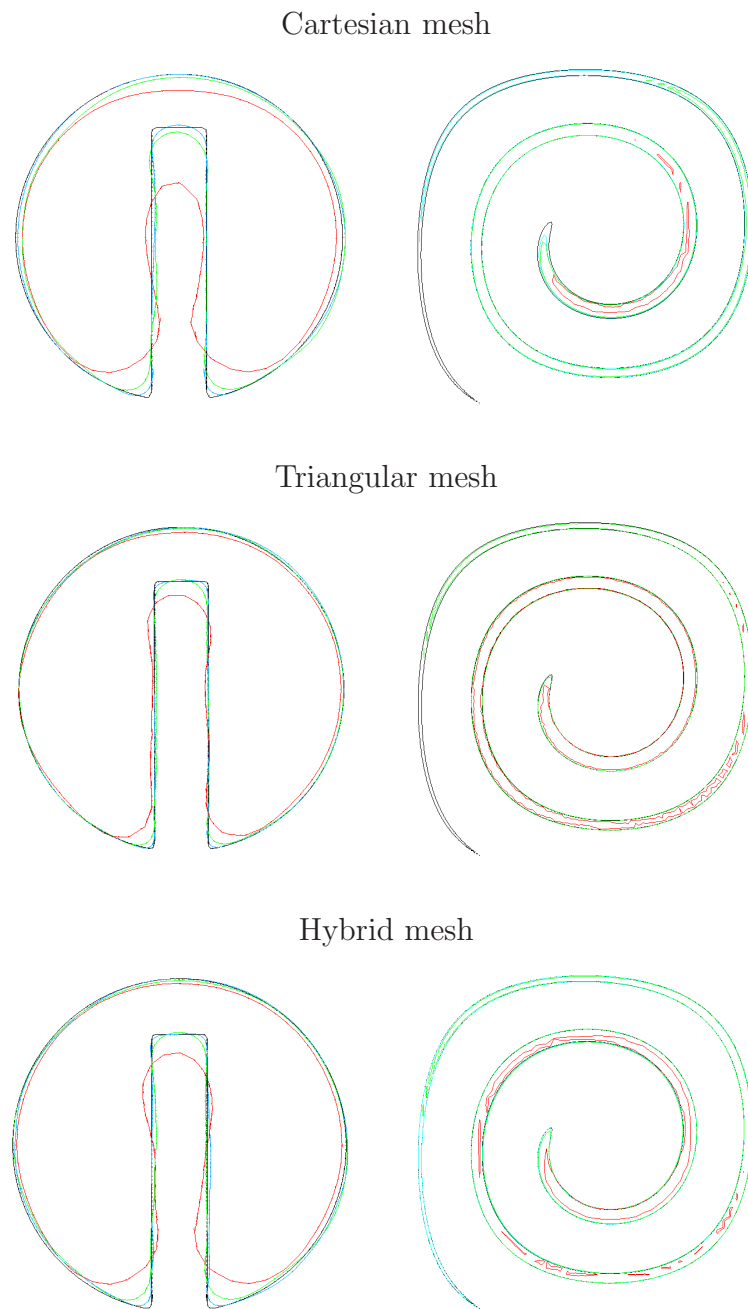


Figure 6.12: Zero level set for the translation of a slotted disk (left column) and the disk in a deformation field (right column) — In black: the reference; in red: solution for $L/64$; in green: solution for $L/128$; in blue: solution for $L/256$

Three-dimensional test case

Three-dimensional meshes For the three-dimensional test case, the meshes have been designed to maintain similar mesh density on the three types of mesh considered (see Figure 6.11, right column, top to bottom): Cartesian mesh (262144 hexes), tetrahedral mesh (250704 tets) and hybrid mesh (234843 elements). With such meshes, we extend our comparison of the performance of the scheme to 3D Cartesian grids and unstructured meshes.

Sphere in a deformation field LeVeque extended the case presented in Section 6.6.1 (see [125]) by considering a sphere in a three-dimensional deformation field given by:

$$\begin{cases} u(x, y, z) = 2 \cdot \sin^2(\pi x) \cdot \sin(2\pi y) \cdot \sin(2\pi z) \\ v(x, y, z) = -\sin(2\pi x) \cdot \sin^2(\pi y) \cdot \sin(2\pi z) \\ w(x, y, z) = -\sin(2\pi x) \cdot \sin(2\pi y) \cdot \sin^2(\pi z) \end{cases} \quad (6.106)$$

In this test, the domain is delimited by the points $(0; 0; 0)$ and $(1; 1; 1)$ and the simulation follows a sphere of radius 0.15, centred on $(0.35; 0.35; 0.35)$. The results obtained with the WENO3 scheme for the sphere in the deformation field are given in Figure 6.13. This figure provides the initial level set field (top row) together with the solution obtained at $t = 0.3125s$ (middle row) and $t = 0.625s$ (bottom row) on the three types of grids considered: Cartesian (left), tetrahedral (middle) and hybrid (right). Therefore, this test case demonstrates the numerical scheme on three-dimensional meshes. It also provides a means to assess the comparative performance of the scheme on various types of 3D mesh.

As for the two-dimensional test cases, the results are comparable for all three types of mesh. The results obtained on the Cartesian mesh are slightly better in terms of smoothness. However, it is worth noting that the Cartesian mesh contains more elements than the tetrahedral mesh (by 5%) and the hybrid mesh (by 12%). It is clear from the results that the present approach works well on fully 3D structured,

unstructured and mixed-element meshes.

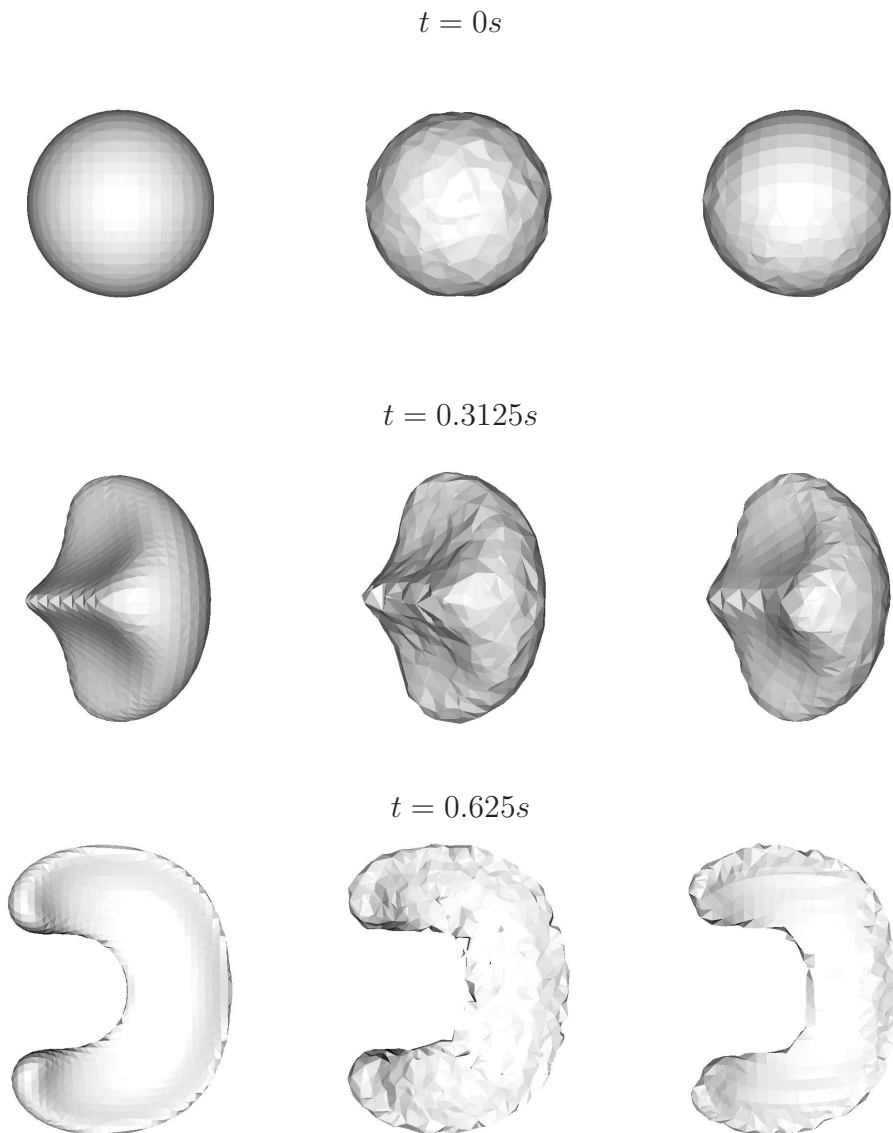


Figure 6.13: Zero level set for the sphere in a deformation field on a Cartesian mesh (left column), a tetrahedral mesh (middle column) and a hybrid mesh (right column)
— Time $t = 0s, 0.3125s, 0.625s$

6.6.2 Numerical convergence study

As in [286], we have chosen to demonstrate the numerical convergence of the WENO scheme on the advection of a sine function. The scalar field is transported in a cube with periodic boundary conditions, such that the solution at $t = 1$ should match the initial field. The linear equation solved and the initial field are given below:

$$\begin{cases} \frac{\partial u}{\partial t} + 2\frac{\partial u}{\partial x} + 2\frac{\partial u}{\partial y} + 2\frac{\partial u}{\partial z} = 0, & (x, y, z) \in [-1; 1]^3 \\ u(x, y, z, 0) = \sin(\pi(x + y + z)) + \sin(2\pi(x + y + z)) \end{cases} \quad (6.107)$$

We have solved this equation using both WENO3 and WENO4 schemes on Cartesian, tetrahedral and hybrid meshes. Three different level of refinement have been considered for this study. Similarly to the 3D meshes in Section 6.6.1, the unstructured meshes have been generated so that — for a given resolution — the different types of mesh have roughly the same number of elements.

The results of the numerical convergence study are presented in tables 6.1 to 6.3. These tables provide the numerical error in the L^1 and L^2 norms and their associated convergence rates calculated using the number of cells N_c in the domain. Introducing the error in the L^p norm \mathcal{E}_{L^p} and the level of grid refinement k , the formula for the order $\mathcal{O}_{L^p}^{(k)}$ reads:

$$\mathcal{O}_{L^p}^{(k)} = \frac{\ln\left(\frac{\mathcal{E}_{L^p}^{(k-1)}}{\mathcal{E}_{L^p}^{(k)}}\right)}{\ln\left(\sqrt[3]{\frac{N_c^{(k)}}{N_c^{(k-1)}}}\right)} \quad (6.108)$$

Although the numerical convergence study is only based on three levels of refinement, the results illustrate that both WENO3 and WENO4 seem to systematically reach a convergence rate significantly higher than the order r of their respective polynomial interpolations. As in [49], we even observe that the order reached by the WENO schemes tends to $r + 1$ regardless of the type of mesh. A more comprehensive convergence study should confirm the trends observed.

In the case of the WENO4 scheme and for the maximum level of resolution considered, the full order of $r + 1$ is not yet reached when calculating the error with

the L^2 norm. However, the convergence rate seems to increase much faster for the WENO4 scheme as the mesh is refined. A similar apparent loss of relative accuracy has been observed by Pilliod and Puckett [178] with high-order schemes on coarse grids. This phenomenon could also be due to the methodology employed to refine the mesh [3]. Indeed, in [49] Dumbser et al. refined their tetrahedral meshes by splitting the elements of the coarse mesh. As this methodology can be cumbersome for mixed-element meshes, it has not been applied in this study.

Table 6.1: Numerical convergence study for the Cartesian meshes — Error and associated order given for both WENO3 and WENO4

| Scheme | Number of cells | \mathcal{E}_{L^1} | \mathcal{O}_{L^1} | \mathcal{E}_{L^2} | \mathcal{O}_{L^2} |
|--------|-----------------|-------------------------|---------------------|-------------------------|---------------------|
| WENO3 | 4096 | 4.8322×10^{-1} | — | 5.321×10^{-1} | — |
| | 32768 | 3.5328×10^{-2} | 3.77 | 3.9349×10^{-2} | 3.76 |
| | 262144 | 2.7226×10^{-3} | 3.70 | 3.1798×10^{-3} | 3.63 |
| WENO4 | 4096 | 5.5752×10^{-1} | — | 6.0496×10^{-1} | — |
| | 32768 | 5.4309×10^{-2} | 3.36 | 6.1376×10^{-2} | 3.3 |
| | 262144 | 1.8432×10^{-3} | 4.88 | 3.2140×10^{-3} | 4.26 |

A plot of the error in the L^2 norm against the normalised computational time is given for both the WENO3 and the WENO4 scheme in figures 6.14 and 6.15 respectively. These graphs illustrate once again that the convergence rates reached by the WENO schemes are independent of the type of grid.

It is interesting to note that figures 6.14 and 6.15 also suggest that — when applying the WENO schemes to the transport of a smooth field — the tetrahedral grids are more computationally efficient than the hybrid and Cartesian grids. This can be explained by the considering the number of side stencils per element and the linear weight given to the side stencils (see Section 6.2). The WENO reconstruction method attributes to an element as many side stencils as it has faces. As a result,

Table 6.2: Numerical convergence study for the tetrahedral meshes — Error and associated order given for both WENO3 and WENO4

| Scheme | Number of cells | \mathcal{E}_{L^1} | \mathcal{O}_{L^1} | \mathcal{E}_{L^2} | \mathcal{O}_{L^2} |
|--------|-----------------|-------------------------|---------------------|-------------------------|---------------------|
| WENO3 | 3511 | 4.5490×10^{-1} | — | 5.1014×10^{-1} | — |
| | 27983 | 3.1000×10^{-2} | 3.88 | 3.5946×10^{-2} | 3.83 |
| | 251906 | 1.7644×10^{-3} | 3.91 | 2.4047×10^{-3} | 3.69 |
| WENO4 | 3511 | 5.2429×10^{-1} | — | 5.8559×10^{-1} | — |
| | 27983 | 5.2578×10^{-2} | 3.32 | 5.8891×10^{-2} | 3.32 |
| | 251906 | 1.8486×10^{-3} | 4.57 | 2.5985×10^{-3} | 4.26 |

for a given number of cells, the Cartesian mesh would involve more side stencils per element than the tetrahedral mesh. The average number of side stencil per element of the hybrid mesh (as constructed in figure 6.11) would lie in between the two. Therefore, the Cartesian mesh requires the largest amount of CPU time followed by the hybrid mesh and then by the tetrahedral mesh. As we chose to give much smaller linear weights to the side stencils than to the central stencil (see Section 6.2), for smooth solutions, the contribution of the additional side stencils to the accuracy of the calculation is not sufficient to offset the additional computational cost incurred.

Table 6.3: Numerical convergence study for the hybrid meshes — Error and associated order given for both WENO3 and WENO4

| Scheme | Number of cells | \mathcal{E}_{L^1} | \mathcal{O}_{L^1} | \mathcal{E}_{L^2} | \mathcal{O}_{L^2} |
|--------|-----------------|-------------------------|---------------------|-------------------------|---------------------|
| WENO3 | 2945 | 5.6115×10^{-1} | — | 6.2401×10^{-1} | — |
| | 27234 | 6.5903×10^{-2} | 2.89 | 8.8502×10^{-2} | 2.63 |
| | 201293 | 4.4836×10^{-3} | 4.03 | 6.1391×10^{-3} | 4.00 |
| WENO4 | 2945 | 5.8506×10^{-1} | — | 6.5202×10^{-1} | — |
| | 27234 | 9.5779×10^{-2} | 2.44 | 1.2464×10^{-1} | 2.23 |
| | 201293 | 5.4088×10^{-3} | 4.31 | 7.7781×10^{-3} | 4.16 |

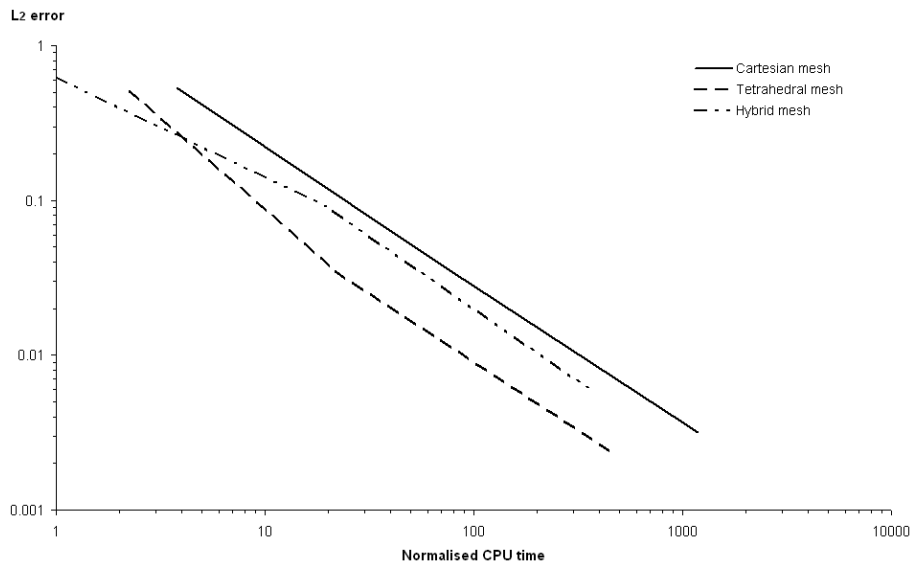


Figure 6.14: L^2 error vs. normalised CPU time for the WENO3 applied to the linear equation

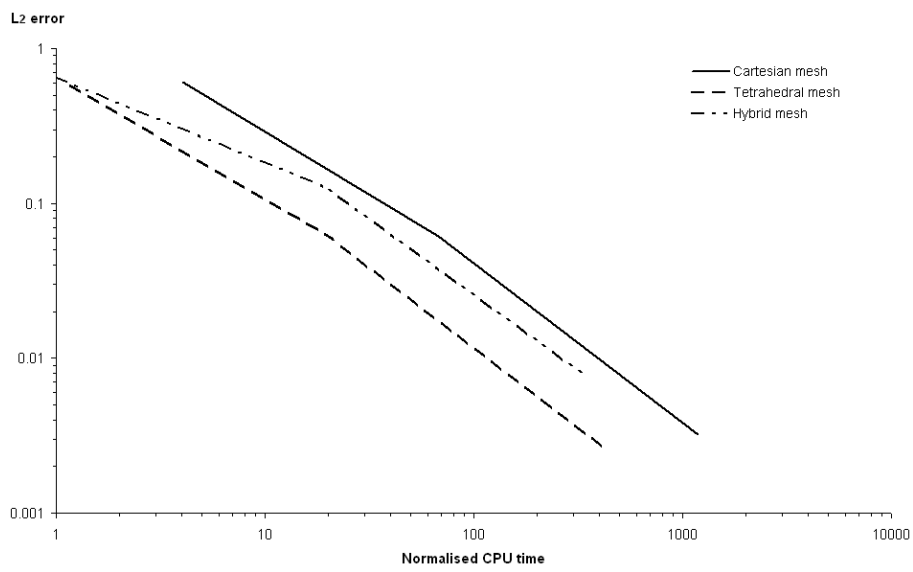


Figure 6.15: L^2 error vs. normalised CPU time for the WENO4 applied to the linear equation

6.6.3 Extension to a non-linear PDE

In order to demonstrate the capability of the WENO scheme on a non-linear PDE, we choose to solve the 3D Burgers' equation on a hybrid mesh of 1.73×10^6 cells. The equation is solved with the WENO3 scheme in a cubical domain with periodic boundary conditions. We use the same initial condition as in [286], such that the shock occurs at time $t = \frac{5}{\pi^2}$. The settings of the simulation are given below:

$$\begin{cases} \frac{\partial u}{\partial t} + \frac{\partial}{\partial x} \left(\frac{u^2}{2} \right) + \frac{\partial}{\partial y} \left(\frac{u^2}{2} \right) + \frac{\partial}{\partial z} \left(\frac{u^2}{2} \right) = 0, & (x, y, z) \in [-3; 3]^3 \\ u(x, y, z, 0) = 0.3 + 0.7 \sin \left(\frac{\pi}{3} (x + y + z) \right) \end{cases} \quad (6.109)$$

The results of the calculation are given in Figure 6.16. A contour plot of the solution on the surface of the domain is shown in the top left corner of Figure 6.16. In the top right corner, we show a contour plot of the solution on the cut at $z = 0$. Finally, in the bottom part of the figure, we compare the numerical solution to the exact solution along the line $x = y$ in the plane $z = 0$.

As intended, the scheme resolves the shock sharply with no trace of oscillatory behaviour. In addition, as suggested by the bottom picture of Figure 6.16, the numerical solution matches very well the exact solution all along the line considered.

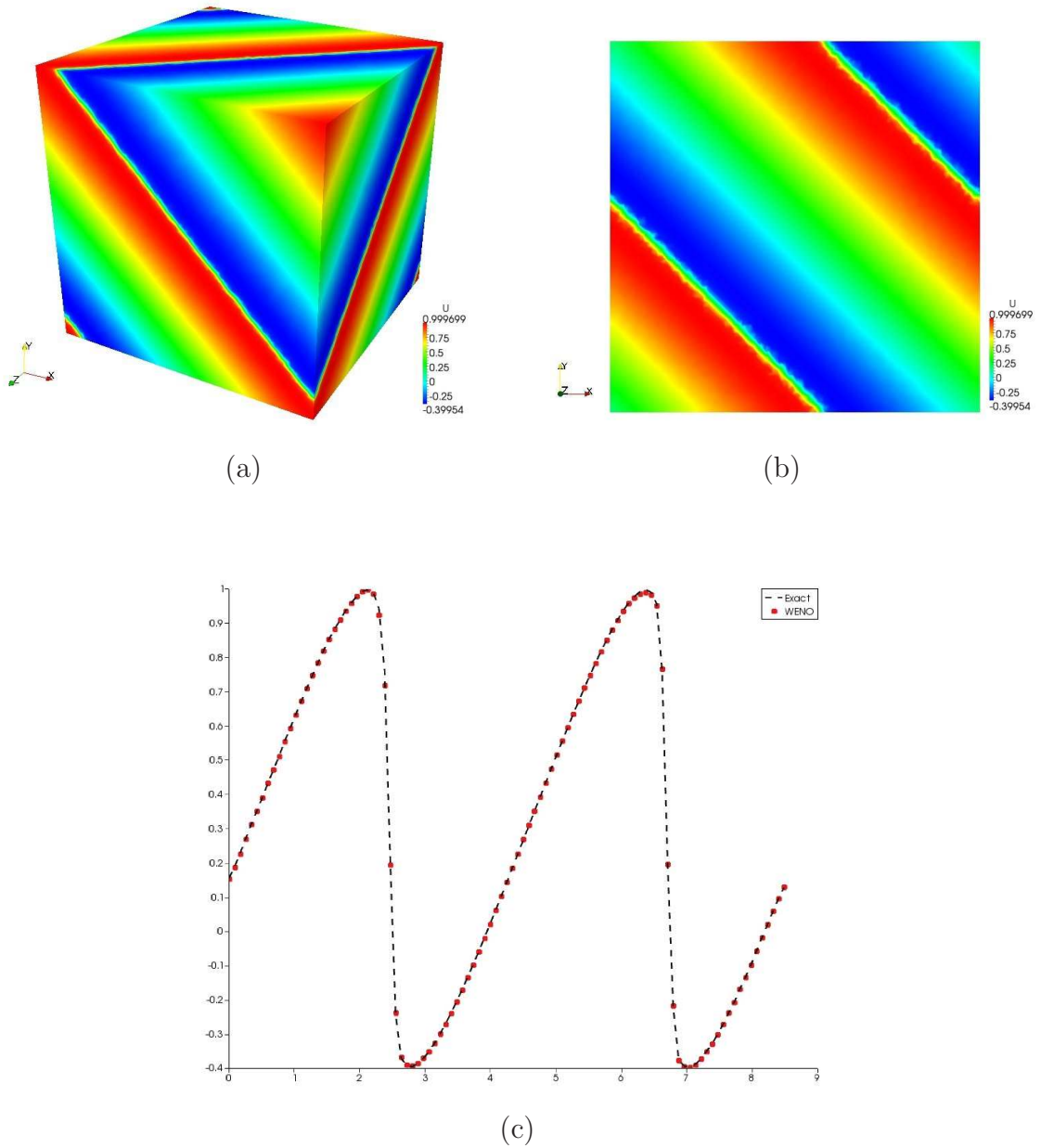


Figure 6.16: Solution for the 3D Burgers' equation on the hybrid mesh at $t = \frac{5}{\pi^2}$ —
 (a) Contour plot of the surface of the domain; (b) contour plot on the cut at $z = 0$;
 (c) exact solution (dashed line) and numerical solution (circles) along the line $x = y$
 in the plane $z = 0$

Chapter 7

Robust conservative level set method

The method developed is based on the conservative level set method (CLS) of Olsson et al. [164]. This is partly because this method generally offers a better trade-off between mass conservation and computational cost than other interface description methods — including state-of-the-art coupled level set-VOF (see for example [152]). In addition, the method can be considered as mature, as it has been successfully applied to the simulation of round liquid jet atomisation by Desjardins et al. [47]. Therefore, the risk associated with implementing and improving such a method is low.

This chapter first provides a brief overview of the robust conservative level set method and then describes the main building blocks of this interface-capturing technique. In particular, the transport of the level set scalar is detailed in Section 7.2, the calculation of the interface normal is presented in Section 7.3 and the flux limiter algorithm is explained in Section 7.4. Finally, the performance of the method is assessed in Section 7.5.

7.1 Overview of the method

As for the CLS method, the level set field transported, φ , can be initialised as a hyperbolic tangent profile of the signed distance function from the interface (ϕ). Introducing ϵ , a coefficient controlling the thickness of the interface, the field φ reads:

$$\varphi = \frac{1}{2} \left(1 + \tanh \left(\frac{\phi}{2\epsilon} \right) \right) \quad (7.1)$$

With such a definition for the level set field φ , the phase boundary is located at $\varphi = \frac{1}{2}$. Although the hyperbolic tangent profile localises the phase transition in the close vicinity of the interface, φ only tends to zero or one, at an infinite distance in, respectively, the gas phase or the liquid phase. To mitigate this issue, the solver as designed also offers the opportunity to set the initial field as the following sine profile:

$$\varphi = \begin{cases} 0 & \text{if } \phi < -\epsilon \\ \frac{1}{2} + \frac{\phi}{2\epsilon} + \frac{1}{2\pi} \sin \left(\frac{\pi\phi}{\epsilon} \right) & \text{if } -\epsilon \leq \phi \leq \epsilon \\ 1 & \text{if } \phi > \epsilon \end{cases} \quad (7.2)$$

The profile given in (7.2), will guarantee that the transition region will not be bigger than 2ϵ .

As in [163, 164], the level set field φ represents the liquid volume fraction and we are taking advantage of the constant thickness of the interface to apply the surface tension via the continuum surface force method. As a result the material properties are given by:

$$\varphi = \begin{cases} \rho = \rho_{gas} + (\rho_{liq} - \rho_{gas}) \varphi \\ \mu = \mu_{gas} + (\mu_{liq} - \mu_{gas}) \varphi \end{cases} \quad (7.3)$$

As the interface is of constant thickness, this method is analogue to a phase field method (see for example [100]). Therefore, it offers the possibility to implement additional equations to describe the interface behaviour on a mesoscopic scale.

The robust conservative level set method (RCLS) that we have developed for finite volume general unstructured meshes operates in three steps:

1. Advection of the liquid volume fraction φ .
2. Re-initialisation of the level set field to maintain the interface thickness constant.
3. Adjustment of the flux of liquid volume fraction to maintain boundedness.

The steps 1 and 2 are described in Section 7.2, and the third step is presented in Section 7.4. The calculation of the interface normal, required by the re-initialisation step, is detailed in Section 7.3.

7.2 Transport of the level set

7.2.1 Mathematical formulation

In the general case, the transport of the liquid volume fraction φ can be expressed by the equation (7.4). Introducing the velocity vector $\mathbf{u} = (u, v, w)$, (7.4) reads:

$$\frac{\partial \varphi}{\partial t} + \mathbf{u} \cdot \nabla \varphi = 0 \quad (7.4)$$

As we are assuming an incompressible framework, (7.4) can be re-written as the following hyperbolic conservation law:

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \mathbf{u}) = 0 \quad (7.5)$$

Even high-order numerics such as the WENO scheme described in Chapter 6 will eventually diffuse the interface. As a result, there is a need to re-initialise the level set profile φ to maintain the interface thickness constant. To achieve that, a compression flux is applied normally to the interface in the transition region. Because the compression flux alone would reduce the interface thickness to nearly zero, leading to a less robust method, a diffusion term is added in the re-initialisation equation. This diffusion term is also applied in the direction normal to the interface, to prevent any tangential diffusion from moving the phase boundary. Introducing

the interface normal, $\hat{\mathbf{n}}$, and the artificial time, τ , along which the equation is solved, the re-initialisation equation reads:

$$\frac{\partial \varphi}{\partial \tau} + \nabla \cdot (\varphi (1 - \varphi) \hat{\mathbf{n}}) = \epsilon \nabla \cdot ((\nabla \varphi \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}}) \quad (7.6)$$

which can be re-written as the following conservation law:

$$\frac{\partial \varphi}{\partial \tau} + \nabla \cdot \left((\varphi (1 - \varphi) - \epsilon (\nabla \varphi \cdot \hat{\mathbf{n}})) \hat{\mathbf{n}} \right) = 0 \quad (7.7)$$

The above equation is solved to steady state, i.e. until the initial level set profile is recovered. All the way through this iterative process, the interface normal, $\hat{\mathbf{n}}$, is kept constant.

7.2.2 Finite volume discretisation

We choose to calculate the numerical fluxes using the exact Riemann solver for both (7.5) and (7.7). As the numerical formulation of the advection equation has already been given in Section 6.4, we will focus on the re-initialisation equation in this sub-section.

Outline of the numerical formulation

In order to solve equation (7.7), we choose to view it as a hyperbolic conservation law in which the gradient $\nabla \varphi$, in the diffusion term, is considered as a constant vector field during each artificial time step m . This gradient is then updated after each iteration of the re-initialisation step. As the normal to the interface is kept constant all the way through the re-initialisation step, the dot product in the diffusion term is constant for each $\Delta \tau$. In the rest of this section, we note this constant scalar field $\varphi_{\hat{\mathbf{n}}}^{(m)}$ for the m^{th} artificial time step, so that we have:

$$(\nabla \varphi \cdot \hat{\mathbf{n}})^{(m)} = \varphi_{\hat{\mathbf{n}}}^{(m)} \quad (7.8)$$

This approach incurs a “splitting error” in artificial time. However, as we are solving the re-initialisation problem to steady state, the accuracy of the artificial

time evolution is not important. Besides, as we use a Runge-Kutta (RK) scheme for the temporal discretisation, we introduce a stronger coupling with the diffusion term as we update the gradient field after each RK sub-step.

In [163], Olsson and Kreiss solve the above equations on Cartesian grids using a total variation diminishing (TVD) scheme with a Superbee limiter. They introduce the diffusion term in the re-initialisation equation to stabilise their CLS method. With our numerical formulation — conservative finite volume WENO scheme and exact Riemann solver — only simulations involving very large density ratios ($\rho_{liq}/\rho_{gas} \approx 1000$) require the diffusion term to be stable.

Alternative numerical treatment

We have also considered treating the equation (7.6) as a hyperbolic conservation law with a source term; the source term being the diffusion term. Such a PDE can be solved by splitting (7.6) into the compression problem (7.9) and the source problem (7.10). This approach also incurs a “splitting error” in artificial time.

The compression problem is given by the following homogeneous non-linear hyperbolic equation:

$$\frac{\partial \varphi}{\partial \tau} + \nabla \cdot (\varphi (1 - \varphi) \hat{\mathbf{n}}) = 0 \quad (7.9)$$

It yields the intermediate scalar field $\varphi^{(1)}$ which serves as initial condition for the following source problem:

$$\frac{\partial \varphi^{(1)}}{\partial \tau} = \epsilon \nabla \cdot ((\nabla \varphi^{(1)} \cdot \hat{\mathbf{n}}) \hat{\mathbf{n}}) \quad (7.10)$$

Regardless of the temporal discretisation employed to solve (7.9) and (7.10), Runge-Kutta or Euler, no significant differences with the other approach were noticed. As the first approach — which includes the diffusion term in the flux calculation — is marginally faster and appears to be more rigorous, it has been implemented in the final version of the code.

Derivation of the numerical flux

Let us consider a discretisation of the computational domain involving conforming elements E_i of volume $|E_i|$ and boundary ∂E_i . Integrating (7.7) over the element E_i leads to:

$$\iiint_{E_i} \frac{\partial \varphi}{\partial \tau} dE_i + \iiint_{E_i} \nabla \cdot \left((\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \hat{\mathbf{n}} \right) dE_i = 0 \quad (7.11)$$

Applying the divergence theorem to (7.11) and simplifying leads to:

$$\frac{d}{d\tau} \bar{\varphi}_i + \frac{1}{|E_i|} \iint_{\partial E_i} (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \hat{\mathbf{n}} \cdot \mathbf{n} d(\partial E_i) = 0 \quad (7.12)$$

where \mathbf{n} is the outward unit vector normal to the surface ∂E_i .

Splitting the integral over the contour of the element ∂E_i into L_i integrals over the faces F_l of E_i , and introducing the outward unit vector normal to F_l : \mathbf{n}_l , we re-write (7.12) as:

$$\frac{d}{d\tau} \bar{\varphi}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \hat{\mathbf{n}} \cdot \mathbf{n}_l d(F_l) = 0 \quad (7.13)$$

Introducing the coordinates of the normal to the interface $\hat{\mathbf{n}} = (\hat{n}_x, \hat{n}_y, \hat{n}_z)$, (7.7) can be re-written in terms of the coordinates (x, y, z) :

$$\begin{aligned} 0 = & \frac{\partial \varphi}{\partial \tau} + \frac{\partial}{\partial x} \left(\hat{n}_x (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \right) + \frac{\partial}{\partial y} \left(\hat{n}_y (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \right) \\ & + \frac{\partial}{\partial z} \left(\hat{n}_z (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \right) \end{aligned} \quad (7.14)$$

Comparing equation (7.14) to equation (6.1) provides the following equalities:

$$\begin{cases} F(\varphi) = \hat{n}_x (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \\ G(\varphi) = \hat{n}_y (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \\ H(\varphi) = \hat{n}_z (\varphi(1-\varphi) - \epsilon \varphi_{\hat{\mathbf{n}}}^{(m)}) \end{cases} \quad (7.15)$$

so that we have:

$$\left\{ \begin{array}{l} \mathbf{A} = (F, G, H) \\ \quad = \left(\hat{n}_x(\varphi(1-\varphi) - \epsilon\overline{\varphi_{\hat{n}}^{(m)}}), \hat{n}_y(\varphi(1-\varphi) - \epsilon\overline{\varphi_{\hat{n}}^{(m)}}), \hat{n}_z(\varphi(1-\varphi) - \epsilon\overline{\varphi_{\hat{n}}^{(m)}}) \right) \\ A_{n_l}(\varphi^-, \varphi^+) = \mathbf{A} \cdot \mathbf{n}_l \end{array} \right. \quad (7.16)$$

The re-initialisation equation can then be expressed in the following finite volume form:

$$\frac{d}{d\tau}\overline{\varphi}_i + \frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} A_{n_l}(\varphi^-, \varphi^+) d(F_l) = 0 \quad (7.17)$$

As for the linear equation (see Section 6.4) and the Burgers' equation (see Section 6.5), simple manipulations of (7.14) demonstrate its rotational invariance according to:

$$\mathbf{A} \cdot \mathbf{n}_l = (F, G, H) \cdot \mathbf{n}_l = n_{lx}F + n_{ly}G + n_{lz}H = \widehat{F} \quad (7.18)$$

where \widehat{F} is the flux vector expressed in the direction n_l , the first axis of the rotated Cartesian frame (n_l, s_l, t_l) . Introducing $\overline{\varphi_{\hat{n}}^{(m)}}$ the face-averaged gradient of the level set in the direction of the normal to the interface, the expression for \widehat{F} reads:

$$\widehat{F} = (\hat{\mathbf{n}} \cdot \mathbf{n}_l) \left(\varphi(1-\varphi) - \epsilon\overline{\varphi_{\hat{n}}^{(m)}} \right) = \hat{n}_{n_l} \left(\varphi(1-\varphi) - \epsilon\overline{\varphi_{\hat{n}}^{(m)}} \right) \quad (7.19)$$

Therefore, the flux across F_l is given by the one-dimensional equation:

$$\frac{\partial \varphi}{\partial \tau} + \frac{\partial \widehat{F}}{\partial n_l} = 0 \quad (7.20)$$

Equation (7.20) leads to the Riemann problem:

$$\left. \begin{array}{l} \text{PDE:} \quad \frac{\partial \varphi}{\partial \tau} + \frac{\partial}{\partial n_l} \left(\hat{n}_{n_l} \left(\varphi(1-\varphi) - \epsilon\overline{\varphi_{\hat{n}}^{(m)}} \right) \right) = 0 \\ \text{IC:} \quad \varphi(n_l, 0) = \varphi_0(n_l) = \begin{cases} \varphi^- & \text{if } n_l < 0 \\ \varphi^+ & \text{if } n_l > 0 \end{cases} \end{array} \right\} \quad (7.21)$$

For the PDE (7.21), the characteristic speed $\lambda(\varphi)$ is given by:

$$\lambda(\varphi) = \frac{d\widehat{F}}{d\varphi} = \hat{n}_{n_l} (1 - 2\varphi) \quad (7.22)$$

Equation (7.21) admits an exact solution:

$$\left. \begin{aligned}
 &\text{If } \lambda(\varphi^-) > \lambda(\varphi^+) : && \varphi(n_l, \tau) = \begin{cases} \varphi^- & \text{if } n_l - S\tau < 0 \\ \varphi^+ & \text{if } n_l - S\tau > 0 \end{cases} \\
 &\text{with: } S = \frac{\Delta \hat{F}}{\Delta \varphi} = \hat{n}_{n_l} (1 - (\varphi^- + \varphi^+)) \\
 &\text{If } \lambda(\varphi^-) \leq \lambda(\varphi^+) : && \begin{cases} \varphi(n_l, \tau) = \varphi^- & \text{if } \frac{n_l}{\tau} \leq \lambda(\varphi^-) \\ \lambda(\varphi) = \frac{n_l}{\tau} & \text{if } \lambda(\varphi^-) < \frac{n_l}{\tau} < \lambda(\varphi^+) \\ \varphi(n_l, \tau) = \varphi^+ & \text{if } \frac{n_l}{\tau} \geq \lambda(\varphi^+) \end{cases}
 \end{aligned} \right\} (7.23)$$

so that the flux A_{n_l} (see (7.16)) across F_l (i.e. at $n_l = 0$ with $t > 0$), reads:

$$\left. \begin{aligned}
 &\text{If } \lambda(\varphi^-) > \lambda(\varphi^+) : && A_{n_l}(\varphi^-, \varphi^+) = \begin{cases} \hat{n}_{n_l} \left(\varphi^- (1 - \varphi^-) - \overline{\epsilon \varphi_{\hat{n}}^{(m)}} \right) & \text{if } S > 0 \\ \hat{n}_{n_l} \left(\varphi^+ (1 - \varphi^+) - \overline{\epsilon \varphi_{\hat{n}}^{(m)}} \right) & \text{if } S < 0 \end{cases} \\
 &\text{with: } S = \hat{n}_{n_l} (1 - (\varphi^- + \varphi^+)) \\
 &\text{If } \lambda(\varphi^-) \leq \lambda(\varphi^+) : && A_{n_l}(\varphi^-, \varphi^+) = \begin{cases} \hat{n}_{n_l} \left(\varphi^- (1 - \varphi^-) - \overline{\epsilon \varphi_{\hat{n}}^{(m)}} \right) & \text{if } 0 \leq \hat{n}_{n_l} (1 - 2\varphi^-) \\ \hat{n}_{n_l} \left(\frac{1}{4} - \overline{\epsilon \varphi_{\hat{n}}^{(m)}} \right) & \text{if } \hat{n}_{n_l} (1 - 2\varphi^-) < 0 < \hat{n}_{n_l} (1 - 2\varphi^+) \\ \hat{n}_{n_l} \left(\varphi^+ (1 - \varphi^+) - \overline{\epsilon \varphi_{\hat{n}}^{(m)}} \right) & \text{if } 0 \geq \hat{n}_{n_l} (1 - 2\varphi^+) \end{cases}
 \end{aligned} \right\} (7.24)$$

As explained in Section 6.3.3, when considering non-linear hyperbolic equations such as (7.8), the Riemann problem has to be solved locally, i.e. for each point of the Gaussian quadrature. The overall flux through the face is then integrated over F_l .

7.2.3 Temporal discretisation

Noting the time variation of φ : $L(\varphi)$, the finite volume form of an hyperbolic conservation law then reads:

$$\begin{aligned} \frac{d}{dt} \bar{\varphi}_i &= -\frac{1}{|E_i|} \sum_{l=1}^{L_i} \iint_{F_l} A_{n_l}(\varphi^-, \varphi^+) d(F_l) \\ &= L(\varphi) \end{aligned} \tag{7.25}$$

Runge-Kutta schemes

We have chosen to use Runge-Kutta schemes for both the advection and the re-initialisation steps. Introducing $\alpha_{i,k}$ and $\beta_{i,k}$, the coefficients of a general Runge-Kutta scheme, the liquid fraction at the i^{th} Runge-Kutta iteration is given by the formula below:

$$\varphi^{(i)} = \sum_{k=0}^{i-1} \alpha_{i,k} \varphi^{(k)} + \beta_{i,k} \Delta t L(\varphi^{(k)}) \tag{7.26}$$

We choose the three-stage, third-order strong-stability preserving (SSP) Runge-Kutta scheme of Shu and Osher [230]: SSP(3, 3). This scheme is widely used in conjunction with WENO schemes because of its stability and accuracy [169, 269]. The coefficients of this scheme are given in the table below.

Table 7.1: Coefficients of the Runge-Kutta scheme SSP(3, 3) of Shu and Osher [230]

| $\alpha_{i,k}$ | | | $\beta_{i,k}$ | | |
|----------------|---------------|---------------|---------------|---------------|---------------|
| 1 | | | 1 | | |
| $\frac{3}{4}$ | $\frac{1}{4}$ | | 0 | $\frac{1}{4}$ | |
| $\frac{1}{3}$ | 0 | $\frac{2}{3}$ | 0 | 0 | $\frac{2}{3}$ |

Other SSP Runge-Kutta schemes were tested (taken from [269]), but the SSP(3, 3) offered the best trade-off between stability, accuracy and computational cost for our numerical formulation.

The solution of the incompressible Navier-Stokes equations in OpenFOAM admits the volumetric flow rate as a variable instead of the velocity. As a result, in order to update the volumetric flow rate after the transport of the liquid volume fraction φ , it is necessary to derive the contribution to the flux for each Runge-Kutta step.

The expression for the liquid volume fraction at the end of the Runge-Kutta scheme (iteration n) can be re-written as:

$$\varphi^{(n)} = K_\alpha \varphi^{(0)} + \sum_{k=0}^{n-1} K_{\beta k} \Delta t L(\varphi^{(k)}) \quad (7.27)$$

where the coefficients K_α and $K_{\beta k}$ are functions of the coefficients of the RK scheme, $\alpha_{i,k}$ and $\beta_{i,k}$, designed such that:

$$K_\alpha = 1 \quad (7.28)$$

$$\sum_{k=0}^{n-1} K_{\beta k} = 1 \quad (7.29)$$

The contribution to the total flux of the k^{th} RK iteration has to be weighted by the coefficient $K_{\beta k}$. We have demonstrated that $K_{\beta k}$ is given by the following formula:

$$K_{\beta k} = \sum_{j=0}^{n-k-1} C_{k+j} \quad (7.30)$$

with:

$$\left. \begin{aligned} j = 0: & \quad C_k = \beta_{i,k} \\ j = 1: & \quad C_{k+1} = \alpha_{i,k+1} \times \beta_{k+1,k} \\ j \geq 2: & \quad C_{k+j} = \alpha_{i,k+j} \times \left(\beta_{k+j,k} + \sum_{m=1}^{j-1} \alpha_{k+j,k+m} \times \frac{C_{m+k}}{\alpha_{i,k+m}} \right) \end{aligned} \right\} \quad (7.31)$$

Stability restrictions on the artificial time step

As the temporal discretisation of the re-initialisation equation is performed with an explicit scheme, it is necessary to consider the stability restrictions associated with

the numerical solution of (7.7). Olsson and Kreiss identify the viscous term in (7.7) as the driver of numerical instabilities and suggest the following condition [163]:

$$\Delta\tau \leq K \frac{(\Delta x)^2}{\epsilon} \quad (7.32)$$

From numerical experiments conducted with a Runge-Kutta scheme, Olsson and Kreiss established that the stability is typically obtained with: $K = \frac{1}{4}$.

7.2.4 Choice of the parameter ϵ

The main parameter of the conservative level set method is the coefficient ϵ that drives the spread of the hyperbolic tangent profile in (7.1). Therefore ϵ effectively controls the thickness of the phase transition.

From (7.1), it is clear that $\varphi \in [0; 1]$. However, as mentioned in Section 7.1, the level set scalar only reaches exactly zero and one at infinity in the gas and liquid phases respectively. As a result, the width of the phase transition is estimated using threshold values for the liquid volume fraction. The hyperbolic tangent profile being symmetrical around $\varphi = 1/2$, we take: $\varphi_{low} = 0.05$ and $\varphi_{high} = 1 - \varphi_{low} = 0.95$ which concentrates 90% of the phase transition. Then, we have the following formula for the interface thickness δ :

$$\begin{aligned} \delta &= 2\epsilon \ln \left(\frac{1 - \varphi_{low}}{\varphi_{low}} \right) \\ &\approx 6\epsilon \end{aligned} \quad (7.33)$$

It is preferable to model the interface as thin as possible, since a sharp interface represents better the reality of the physics and involve less smearing of the material properties and the surface tension. A small ϵ is also desirable to minimize the effect of the re-initialisation step. Indeed, although Olsson et al. proved that the re-initialisation step would not move the phase boundary ($\varphi = 0.5$) in 1D [164], this property has yet to be demonstrated for two and three-dimensional domains. However, if $\epsilon \rightarrow 0$ it is certain that the re-initialisation step will not move the 0.5-contour of φ .

Nevertheless, the interface needs to have a minimal thickness, so that the gradient of φ and the interface normal are accurately calculated. A minimum value for ϵ also results from a “stability study” of the 1D re-initialisation equation. Olsson et al. performed this study for a finite element method in [164]. We will apply below the same methodology for finite volume methods. Let us re-write the equation (7.11) in the one-dimensional case:

$$\int_{E_i} \frac{\partial \varphi}{\partial \tau} dE_i + \int_{E_i} \nabla \cdot \left((\varphi(1-\varphi) - \epsilon \varphi \hat{n}) \hat{n} \right) dE_i = 0 \quad (7.34)$$

As we are solving the re-initialisation equation to steady-state, we are effectively seeking φ^n that satisfies the following equality in 1D:

$$\int_{x_i}^{x_{i+1}} \varphi^n (1 - \varphi^n) dx = \epsilon \int_{x_i}^{x_{i+1}} \varphi_x^n dx \quad (7.35)$$

where x_i and x_{i+1} are the borders of the 1D element E_i along the coordinate x .

We note: $h_i = x_{i+1} - x_i$, $\varphi_i = \varphi^n(x_i)$ and $\varphi_{i+1} = \varphi^n(x_{i+1})$. In a finite volume framework, the scalar φ is constant and equal to φ_i over each element E_i centred on x_i . Assuming the x_i of the 1D mesh equidistant, then equation (7.35) will read after integration:

$$\frac{h_i}{2} \left(\varphi_i (1 - \varphi_i) + \varphi_{i+1} (1 - \varphi_{i+1}) \right) = \epsilon (\varphi_{i+1} - \varphi_i) \quad (7.36)$$

In the case where $\varphi_i \approx \varphi_{i+1} \approx 0$, linearising the above equality around zero leads to:

$$\varphi_{i+1} \approx \frac{\epsilon + h_i/2}{\epsilon - h_i/2} \varphi_i \quad (7.37)$$

As a result, if $\epsilon < h_i/2$, φ_i and φ_{i+1} will have opposite signs and φ^n may oscillate. A similar analysis when $\varphi_i \approx \varphi_{i+1} \approx 1$ leads to:

$$\varphi_{i+1} \approx \frac{\epsilon - h_i/2}{\epsilon + h_i/2} \varphi_i + \frac{h_i}{\epsilon + h_i/2} \quad (7.38)$$

Therefore, the same constraint appears in the general case when linearising around unity. However, instability is less likely as the formula relating φ_{i+1} to φ_i involves a second term which is always positive.

In conclusion of this stability study, in order to maintain a stable solution of the re-initialisation equation, it is necessary to choose:

$$\epsilon \geq \frac{1}{2}\Delta x \quad (7.39)$$

On unstructured grids, as the mesh size varies, it is safer to take $\epsilon \geq \frac{1}{2} \max(\Delta x)$.

After Olsson and Kreiss, we choose to take ϵ proportional to the grid size:

$$\epsilon = C\Delta x \quad (7.40)$$

The consequence of that choice is twofold:

- The equations solved for the transport of the liquid volume fraction change as the grid is refined.
- As Δx decreases, the profile of the conservative level set in the phase transition region is resolved using the same number of cells.

Olsson and Kreiss recommended to use $\epsilon = C(\Delta x)^{1-d}$ with $d \in [0; 1]$ when convergence could not be obtained. In particular, they used $d = 0.1$ in one of the numerical tests of [163].

7.2.5 Initialisation of the conservative level set field

On simple test cases, the signed distance function from the interface, ϕ , may be calculated analytically so that the initial conservative level set field $\varphi^{(0)}$ can be easily derived by applying the formula (7.1) in each cell of the mesh.

If the field cannot be calculated analytically, one way to obtain $\varphi^{(0)}$ is to compute ϕ first by solving the re-distancing equation (3.27) to steady state [32, 242, 244] or by using Sethian's fast marching method [229]. However, none of these methods exists for general polyhedral meshes.

Another option, described by Olsson and Kreiss in [163], is to initialise φ as a VOF field — setting 0 in the cells containing gas and 1 in the cells containing liquid — and then to solve the re-initialisation equation (7.7) to steady state. This procedure will produce an initial conservative level set field of reasonable quality. For a generic distribution of liquid on a general polyhedral mesh, this is the only viable method.

7.3 Calculation of the interface normal

7.3.1 Mathematical formulation

The face-averaged gradient, $\overline{\nabla\varphi_{\mathbf{x}}}$, is required for the calculation of the interface normal, the non-linear flux and the surface tension forces. Let us recall the definition of $\overline{\nabla\varphi_{\mathbf{x}}}$ below:

$$\overline{\nabla\varphi_{\mathbf{x}}} = \frac{1}{|F_l|} \iint_{F_l} \nabla\varphi_{\mathbf{x}} \, d(F_l) \quad (7.41)$$

Desjardins observed in [46] that the quality of the gradient field calculation was important to avoid spurious oscillations. To resolve this issue, we have chosen to take advantage of the polynomial reconstruction of the scalar field φ . As the polynomial reconstruction is performed in a reference space $\boldsymbol{\xi} = (\xi, \eta, \zeta)$ — where scaling effects do not apply (see Section 6.3.1) — the smeared out liquid volume fraction φ is approximated by a WENO polynomial according to the following formula (see Chapter 6):

$$\varphi_{WENO}(\xi, \eta, \zeta) = \bar{\varphi}_0 + \sum_{k=1}^K \tilde{a}_k \phi_k(\xi, \eta, \zeta) \quad (7.42)$$

where the \tilde{a}_k are the degrees of freedom of the WENO polynomial reconstruction and the $\phi_k(\xi, \eta, \zeta)$ are the basis functions of the polynomial reconstruction (see Section 6.3.2).

This polynomial — already calculated by the high-order WENO scheme for the linear flux of the advection equation — embeds both the essentially non-oscillatory

characteristic (required to avoid the spurious oscillations mentioned by Desjardins) and high-order approximation of the gradient.

As the gradient is needed in the physical space, the Jacobian of the mapping transformation, \mathcal{J} , has to be introduced in the calculation, according to:

$$\nabla\varphi_{\mathbf{x}} = (\mathcal{J}^{-1})^T \nabla\varphi_{\boldsymbol{\xi}} \quad (7.43)$$

Based on equations (7.41), (7.42) and (7.43) and recalling that F'_l is the face F_l in the mapped space, the following expression can be derived for the face-averaged gradient:

$$\overline{\nabla\varphi_{\mathbf{x}}} = \sum_{k=1}^K \tilde{a}_k \underbrace{\left(\frac{1}{|F'_l|} (\mathcal{J}^{-1})^T \mathbf{v}_k \right)}_{\text{pre-computed}} \quad (7.44)$$

where the vector \mathbf{v}_k reads:

$$\mathbf{v}_k = \begin{pmatrix} A_k \iint_{F'_l} \xi^{(A_k-1)} \eta^{B_k} \zeta^{C_k} d(F'_l) \\ B_k \iint_{F'_l} \xi^{A_k} \eta^{(B_k-1)} \zeta^{C_k} d(F'_l) \\ C_k \iint_{F'_l} \xi^{A_k} \eta^{B_k} \zeta^{(C_k-1)} d(F'_l) \end{pmatrix} \quad (7.45)$$

The added cost for the above calculation is fairly small as the the degrees of freedom are computed for the flux determination and since the terms multiplying the \tilde{a}_k in (7.44) are precomputed.

Besides, the integrals of the monomials in (7.45) are simple combinations of the volume integrals of the monomial and the surface integrals of the basis functions. Let us recall the definition of the basis functions:

$$\phi_k = \psi_k - \frac{1}{|E'_i|} \iiint_{E'_i} \psi_k d(E'_i) \quad (7.46)$$

with: $\{\psi_k\} = \xi, \eta, \zeta, \xi^2, \xi \cdot \eta, \dots, \zeta^r \quad k = 1, \dots, K$.

It results that the integrals of the monomials, ψ_k , involved in \mathbf{v}_k can be calculated from:

$$\iint_{F'_l} \psi_k \, d(F'_l) = \underbrace{\iint_{F'_l} \phi_k \, d(F'_l)}_{\text{pre-computed for the flux}} - \frac{|F'_l|}{|E'_i|} \underbrace{\iiint_{E'_i} \psi_k \, d(E'_i)}_{\text{pre-computed for } A_{jk}} \quad (7.47)$$

As a result, even the pre-processing step is not expanded significantly as both of the integrations on the r.h.s. of (7.47) have already been performed: the first one for the flux calculation and the second one for the polynomial reconstruction.

7.3.2 Numerical tests

The performance of the high-order approximation of the gradient has been tested on a triangular grid. The conservative level set has been set such that the contours of φ follow a sine function (see Figure 7.1). Besides, various interface thicknesses have been considered: in this study the CLS parameter ϵ takes values in the range: $[0.5; 2]$.

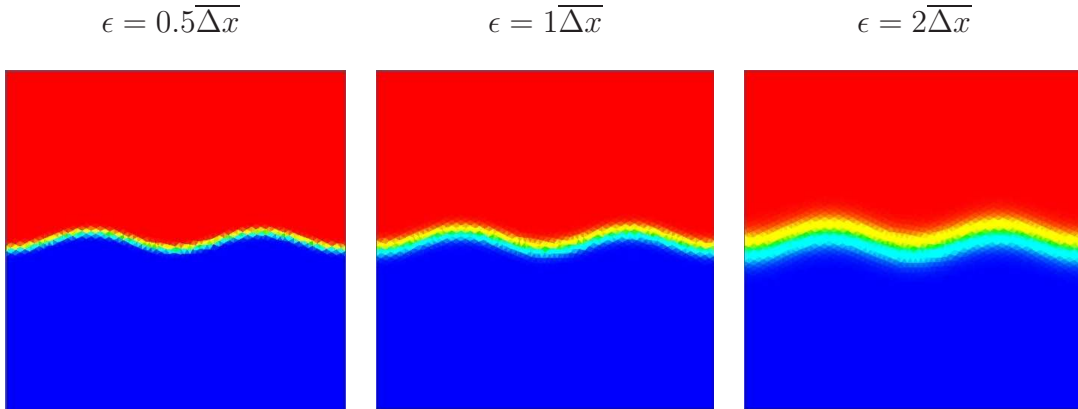


Figure 7.1: Contour plots of the CLS field for gradient performance tests — Interface thicknesses: $\epsilon = 0.5\overline{\Delta x}$, $1\overline{\Delta x}$, $2\overline{\Delta x}$

The performance of the high-order gradient approximation has been measured against a typical low-order gradient calculation that provides the face-averaged gra-

dient $(\overline{\nabla\varphi})_l$ for the face F_l . The determination of $(\overline{\nabla\varphi})_l$ is based on the cell-averaged gradient of the elements E_i and E_j separated by the face F_l . Introducing the face-interpolated scalar φ_l and the centres \mathbf{C}_l , \mathbf{C}_i and \mathbf{C}_j of respectively F_l , E_i and E_j , the application of Gauss' theorem to $(\overline{\nabla\varphi})_i$ leads to:

$$(\overline{\nabla\varphi})_i = \frac{1}{|E_i|} \sum_{l=1}^{L_i} \varphi_l |F_l| \mathbf{n}_l \quad \text{where:} \quad \begin{cases} \varphi_l = f_x \varphi_i + (1 - f_x) \varphi_j \\ f_x = \frac{|\mathbf{C}_l - \mathbf{C}_j|}{|\mathbf{C}_l - \mathbf{C}_j| + |\mathbf{C}_l - \mathbf{C}_i|} \end{cases} \quad (7.48)$$

Applying the same face-interpolation method to the gradient, the face-averaged gradient then reads:

$$(\overline{\nabla\varphi})_l = f_x (\overline{\nabla\varphi})_i + (1 - f_x) (\overline{\nabla\varphi})_j \quad (7.49)$$

The results of the computations are shown in Figure 7.2 and 7.3. Figure 7.2 compares the low and high-order approaches in terms of gradient magnitude for three different thicknesses: $\epsilon = 0.5$, $\epsilon = 1$ and $\epsilon = 2$. Similarly, in Figure 7.3, the performance of the low and high-order calculations are compared for the horizontal component (direction x) of the gradient. The component of the gradient in the vertical direction is not shown here as it closely matches the gradient magnitude.

The contour plots in Figure 7.2 and 7.3 demonstrate the superior performance of the high-order gradient calculation. In particular, in Figure 7.2, the contour plots clearly show that the gradient is sharper with the high-order approach. Besides, the convoluted sine shape of the interface does not affect the quality of the high-order gradient which remains homogeneous all along the contours of φ . This is not the case for the low-order approach. Indeed, regardless of the interface thickness, the contour plots of low-order gradient magnitude presents several irregularities along the phase boundary at $\varphi = \frac{1}{2}$.

In Figure 7.3, the contour plots of the x-component of the low-order gradient also show some irregularities near the phase boundary, where the variation of φ is the

steepest. This suggests a tendency for the low-order method to generate spurious oscillations. In addition, the low-order method is affected by the type of mesh. This is particularly obvious for the thickest interface. As expected, the high-order gradient is free of spurious oscillations and copes remarkably well with unstructured meshes.

In conclusion, the high-order gradient calculation offers a significant gain in accuracy and is free from spurious oscillations.

NB: By default, the modelling capability calculates the gradient of φ using the high-order approach. However, the gain in performance offered by this method may vary, depending on the test case considered. For example, no significant improvements were observed for the Rayleigh-Taylor instability test case. To offer more flexibility to the user, the code leaves the option to revert back to the low-order approach to save computational time.

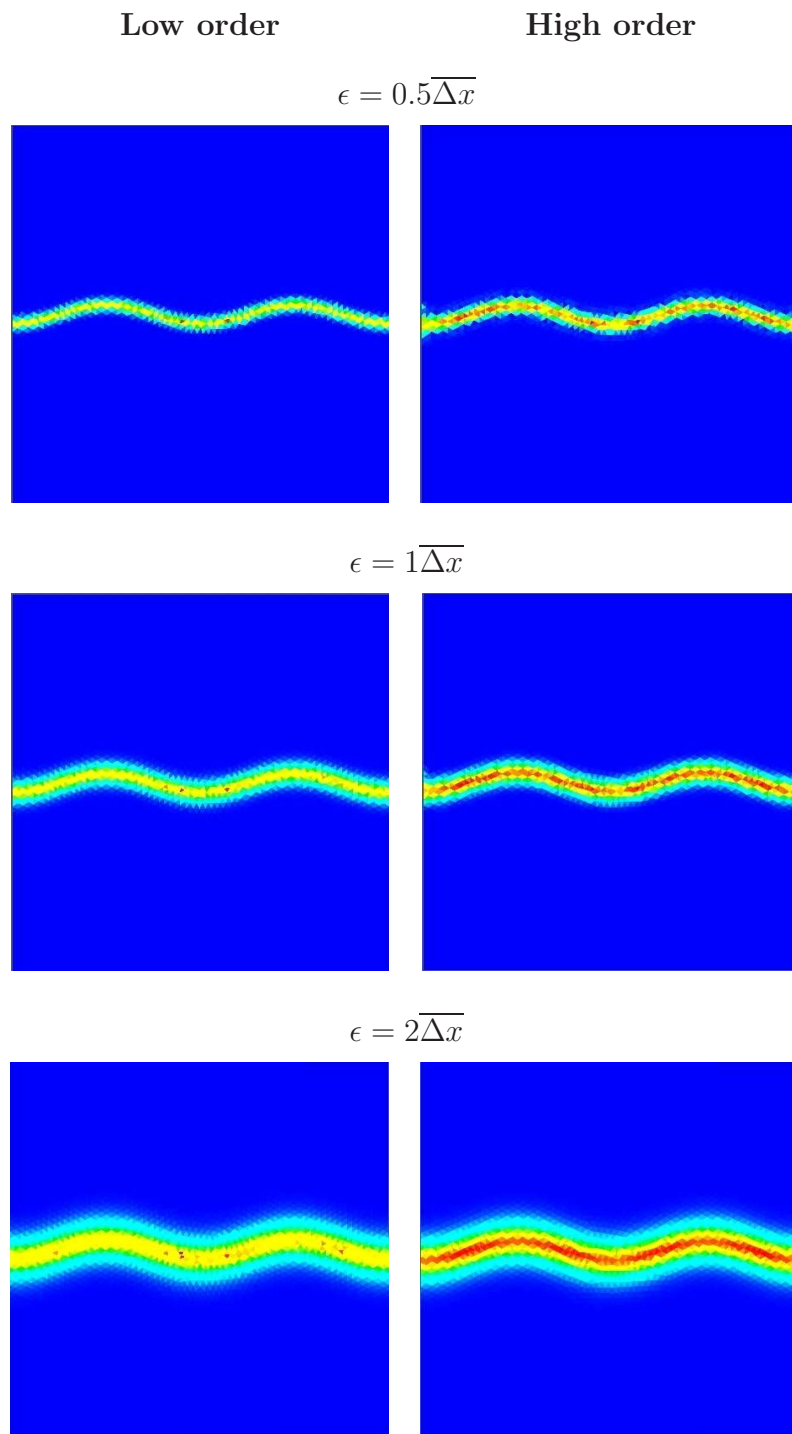


Figure 7.2: Comparative performance of the gradient calculation: low order vs. high order — Gradient magnitudes for interface thicknesses: $\epsilon = 0.5\overline{\Delta x}$, $1\overline{\Delta x}$, $2\overline{\Delta x}$

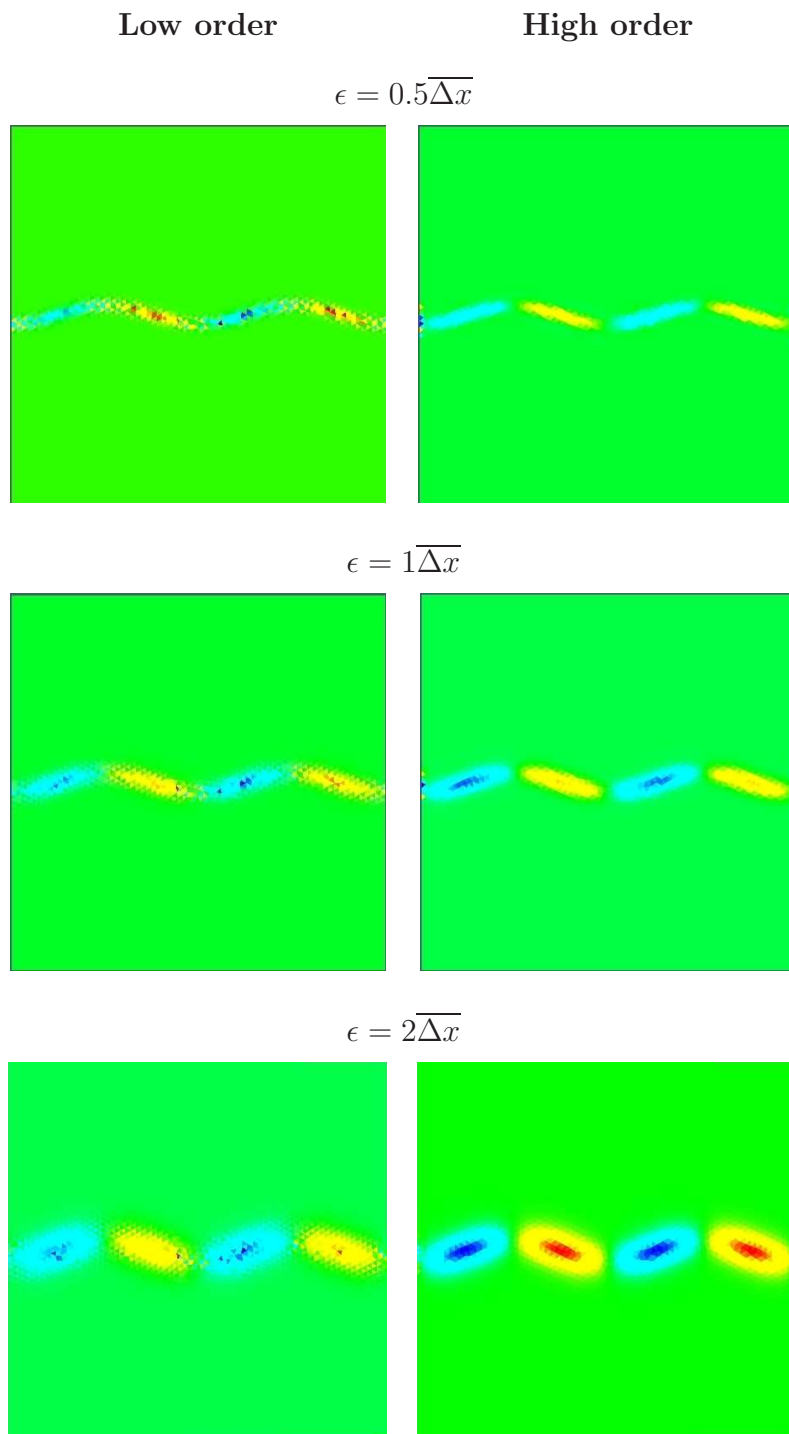


Figure 7.3: Comparative performance of the gradient calculation: low order vs. high order — Gradient in the direction x for interface thicknesses: $\epsilon = 0.5\overline{\Delta x}$, $1\overline{\Delta x}$, $2\overline{\Delta x}$

7.4 Multidimensional universal limiter with explicit solution

As noted in [46, 269], WENO schemes are not necessarily total variation bounded (TVB), even when coupled with a total variation diminishing (TVD) Runge-Kutta time integration. However, as our conservative level set field represents the liquid volume fraction in the domain, non-physical values of φ such that $\varphi < 0$ or $\varphi > 1$ cannot be tolerated. Indeed, such non-physical values may worsen over millions of time steps and eventually the density, calculated from φ , may end up negative in some cells. This would then crash the solver employed to solve the Navier-Stokes incompressible equations.

As a result, we decided to use the multidimensional universal limiter with explicit solution (MULES) of Weller [166], used by default in OpenFOAM to maintain the boundedness of the VOF field. It is worth mentioning that the addition of such a flux limiter algorithm to the RCLS will reduce the overall order of accuracy of the interface-capturing method [3].

To our knowledge, Weller has not presented his algorithm in any publication. The only available description of the method is given by Massé in [148]. This section presents the MULES method based on Massé's thesis and on OpenFOAM's routine for MULES.

7.4.1 Overview of the method

MULES calculates a limited flux through the face F_l , $\widehat{F}_{L,l}$, for the liquid volume fraction φ such that for each cell E_i , the conservative level set at time t_{n+1} , $\varphi_i^{(n+1)}$, remains bounded by the minimal and maximal values of the solution in its neighbouring cells E_{j_l} at the previous time step t_n :

$$\min_{j_l} \left(\varphi_{j_l}^{(n)} \right) \leq \varphi_i^{(n+1)} \leq \max_{j_l} \left(\varphi_{j_l}^{(n)} \right) \quad j_l = 1, \dots, L_i \quad (7.50)$$

Let us note $\varphi_{i,min}^{(n+1)}$ and $\varphi_{i,max}^{(n+1)}$, respectively the minimum and the maximum admissible values for $\varphi_i^{(n+1)}$ in (7.50). To ensure that no non-physical values will be propagated, both $\varphi_{i,min}^{(n+1)}$ and $\varphi_{i,max}^{(n+1)}$ are then clipped between 0 and 1, using the following clipping operator:

$$\text{clip}(x) = \max(\min(x, 1), 0) \quad (7.51)$$

MULES follows the basic principles of the flux-corrected transport (FCT) of Boris and Book [22–24], in the format described by Zalesak [284]. In order to satisfy the boundedness criterion (7.50), MULES calculates the limited flux $\widehat{F}_{L,l}$ through F_l face by face by taking the weighted average of two fluxes for φ :

- The flux computed with a first-order upwind scheme for the advection equation only: $\widehat{F}_{U,l}$. This flux leads to a bounded solution but is diffusive.
- The sum of the fluxes computed with the high-order scheme (see Chapter 6), $\widehat{F}_{HO,l}$, for both the advection equation and the re-initialisation equation. This flux leads to a high-order accurate solution that is not systematically bounded.

As the weighting varies from face to face, the FCT computation of the limited flux is non linear.

Introducing the limiter factor λ_l — calculated for each face F_l of the mesh — the expression for the limited flux then reads:

$$\widehat{F}_{L,l} = (1 - \lambda_l) \widehat{F}_{U,l} + \lambda_l \widehat{F}_{HO,l} \quad (7.52)$$

The flux of liquid volume fraction \widehat{F}_l through a given face F_l represents the amount of liquid going through F_l during a given time step Δt . Although not numerically calculated as such, the flux is physically identical to:

$$\widehat{F}_l = |F_l| (\mathbf{U}_l \cdot \mathbf{n}_l) \varphi$$

where \mathbf{U}_l is the face-averaged velocity vector over F_l and \mathbf{n}_l is the outward face normal of the cell considered E_i . As a result, the flux is positive when the liquid is

exiting the cell E_i and negative when entering it. Of course, for two neighbouring cells E_i and E_j , separated by the face F_l , the flux through F_l is equal in magnitude and of opposite sign for each cell.

Introducing the correction flux $\widehat{F}_{C,l}$ defined as the difference between $\widehat{F}_{HO,l}$ and $\widehat{F}_{U,l}$:

$$\widehat{F}_{C,l} = \widehat{F}_{HO,l} - \widehat{F}_{U,l} \quad (7.53)$$

The expression for the limited flux (7.52) can then be re-written as:

$$\widehat{F}_{L,l} = \widehat{F}_{U,l} + \lambda_l \widehat{F}_{C,l} \quad (7.54)$$

Once the limiter factor computed and the limited flux calculated according to (7.54), the scalar field φ is updated using an Euler time integration, such that:

$$\varphi_i^{(n+1)} = \varphi_i^{(n)} - \frac{\Delta t}{|E_i|} \sum_{l=1}^{L_i} \widehat{F}_{L,l} \quad (7.55)$$

7.4.2 Determination of the limiter factor

The limiter factor λ_l is calculated iteratively such that the boundedness condition (7.50) is satisfied. λ_l directly derives from the re-formulation of the condition (7.50) in terms of fluxes. The bounds for the liquid volume fraction in E_i at time t_{n+1} correspond, in terms of fluxes, to the maximum temporal variation of liquid volume in E_i , $\left(\frac{\Delta V_{liq}}{\Delta t}\right)_i$:

- The minimum $\varphi_{i,min}^{(n+1)}$ corresponds to the maximum decrease of liquid volume in E_i .
- The maximum $\varphi_{i,max}^{(n+1)}$ corresponds to the maximum increase of liquid volume in E_i .

These temporal variations — homogeneous to a flux — read respectively:

$$\left\{ \begin{array}{l} \left(\frac{\Delta V_{liq}}{\Delta t} \right)_i^{out_{max}} = \frac{\varphi_i^{(n)} - \varphi_{i,min}^{(n+1)}}{\Delta t} |E_i| \\ \left(\frac{\Delta V_{liq}}{\Delta t} \right)_i^{in_{max}} = \frac{\varphi_{i,max}^{(n+1)} - \varphi_i^{(n)}}{\Delta t} |E_i| \end{array} \right. \quad (7.56)$$

so that fluxes limited by the temporal variations in (7.56), will lead to a bounded solution for φ .

Weller's algorithm searches iteratively the maximum value of λ_l that satisfies the boundedness criterion. Starting from $\lambda_l^{(0)} = 1$ — i.e. $\widehat{F}_{L,l} = \widehat{F}_{HO,l}$ — which leads to a high-order accurate solution, the algorithm progressively increases the contribution of $\widehat{F}_{U,l}$ to the flux by reducing λ_l .

To achieve that, the fundamental principle of MULES is to split the sum of correction fluxes for a given cell E_i into the sum of outflow and the sum of inflow correction fluxes. Then, in each cell, and at a given iteration k of the algorithm, an average limiter factor $(\lambda_i^{out})^{(k)}$ is defined for all the faces that support an outflow correction flux and another one $(\lambda_i^{in})^{(k)}$ is defined for all the faces that support an inflow correction flux.

These average limiter factors are introduced in the boundedness conditions expressed in terms of fluxes. These conditions then read:

$$\left(\frac{\Delta V_{liq}}{\Delta t} \right)_i^{out_{max}} = (\lambda_i^{out})^{(k)} \sum_{l,out} \widehat{F}_{C,l} - \sum_{l,in} \lambda_l^{(k-1)} \widehat{F}_{C,l} + \sum_{l=1}^{L_i} \widehat{F}_{U,l} \quad (7.57)$$

$$\left(\frac{\Delta V_{liq}}{\Delta t} \right)_i^{in_{max}} = (\lambda_i^{in})^{(k)} \sum_{l,in} \widehat{F}_{C,l} - \sum_{l,out} \lambda_l^{(k-1)} \widehat{F}_{C,l} - \sum_{l=1}^{L_i} \widehat{F}_{U,l} \quad (7.58)$$

where the sums of outflow and inflow fluxes in E_i are calculated the following way for a given flux \widehat{F}_l :

$$\left\{ \begin{array}{l} \sum_{l,out} \widehat{F}_l = \sum_{l=1}^{L_i} \max(\widehat{F}_l, 0) \\ \sum_{l,in} \widehat{F}_l = \sum_{l=1}^{L_i} \min(\widehat{F}_l, 0) \end{array} \right. \quad (7.59)$$

As a flux exiting a cell — and therefore decreasing the amount of liquid volume in the cell — is positive, the sum of upwind fluxes $\widehat{F}_{U,l}$ on all the faces of E_i is then added on the r.h.s. of (7.57). Similarly, the sum of upwind fluxes is subtracted on the r.h.s. of (7.58).

The average limiter factors $(\lambda_i^{out})^{(k)}$ and $(\lambda_i^{in})^{(k)}$ are then calculated from (7.57) and (7.58) respectively.

As can be seen in the boundedness condition relating to the maximum decrease in liquid volume (7.57), $(\lambda_i^{out})^{(k)}$ weights the sum of outflow correction fluxes while the limiter factor obtained at the previous iteration $\lambda_l^{(k-1)}$ weights individually each of the corresponding inflow correction fluxes. Similarly, in the boundedness condition relating to the maximum increase in liquid volume (7.58), $(\lambda_i^{in})^{(k)}$ weights the sum of inflow correction fluxes while the limiter factor obtained at the previous iteration $\lambda_l^{(k-1)}$ weights individually each of the corresponding outflow correction fluxes.

Then, the algorithm takes as limiter factor for the face F_l at the current iteration k , $\lambda_l^{(k)}$, the minimum of three values:

- The relevant average limiter factor in one of the adjacent cells E_i : $(\lambda_i^{out})^{(k)}$ if the correction flux through F_l is exiting E_i , $(\lambda_i^{in})^{(k)}$ otherwise.
- The relevant average limiter factor in the other adjacent cell E_j : $(\lambda_j^{in})^{(k)}$ if the correction flux through F_l is entering E_j , $(\lambda_j^{out})^{(k)}$ otherwise.
- The limiter factor at the previous iteration: $\lambda_l^{(k-1)}$.

The algorithm iterates m times — with m specified by the user — to produce a final limiter factor $\lambda_l^{(m)}$ for each face F_l of the mesh.

7.5 Performance of the method

This section compares the performance of the RCLS with established multiphase numerical methods:

- The VOF method of OpenFOAM: *interFoam*
- The accurate conservative level set (ACLS) method of Desjardins et al. [46, 47]

For this comparative study, the RCLS transports the scalar field φ using a WENO3 scheme, the CLS coefficient is set to $\epsilon = 0.5\Delta x$ and the re-initialisation of the hyperbolic tangent profile is only performed every five time steps (see Section 9.2.2).

Comparison with *interFoam*

The relative performance of the transport algorithms is assessed on the following test cases (defined in Section 4.2.1): Zalesak’s slotted disk (on a 128^2 Cartesian mesh) and the disk in a deformation field (on a 256^2 Cartesian mesh).

As can be seen in Figure 7.4, the RCLS method clearly outperforms *interFoam* on both test cases. In particular, the ligament predicted by our transport algorithm is longer and the interface is free from non-physical wiggles.

Comparison with ACLS

In order to compare the method of Desjardins et al. with the RCLS, we have performed the computation of the disk in a deformation field as set in [47]. In this publication, the deformation field defined by the authors differs from the one given

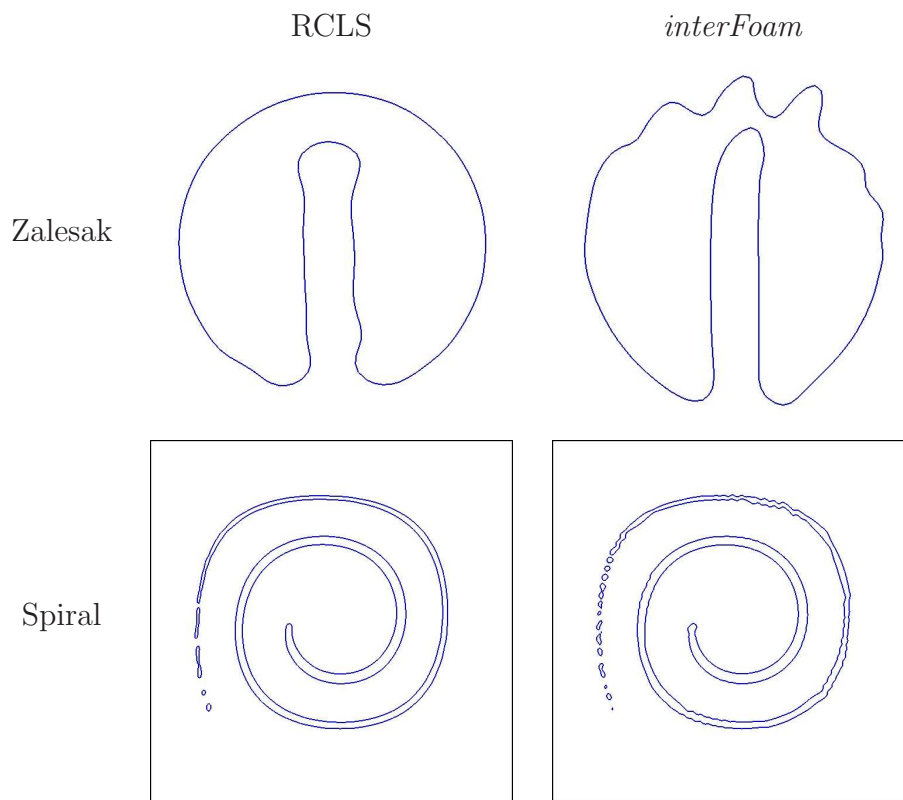


Figure 7.4: Performance of the transport algorithms of *interFoam* and the RCLS method — Results for Zalesak’s slotted disk (Zalesak) and the disk in a deformation field (Spiral)

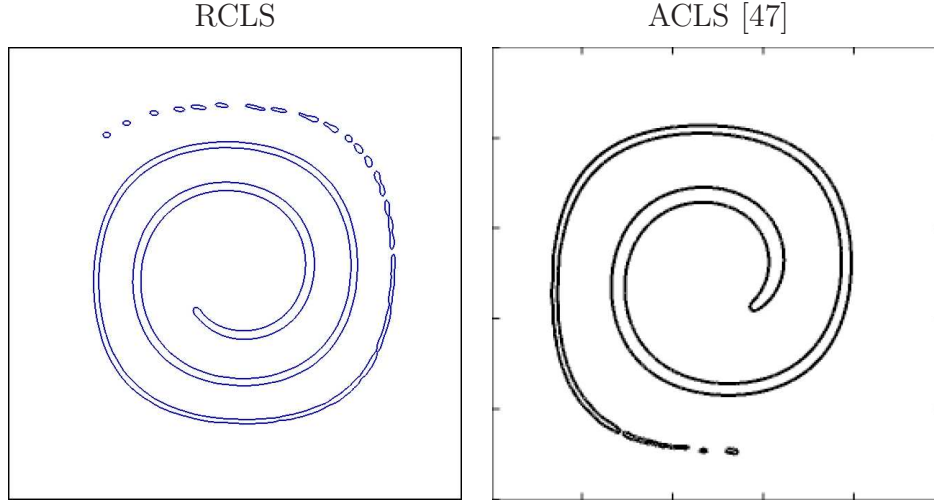


Figure 7.5: Performance of the transport algorithms of the ACLS and the RCLS methods — Results for the disk in a deformation field (as set in [47])

in Section 4.2.1. As in [47], we set the stream function to:

$$\psi = \frac{1}{\pi} \sin^2(\pi x) \cos^2(\pi x) \cos\left(\frac{\pi t}{T}\right) \quad (7.60)$$

with $T = 8s$

The resulting velocity vector \mathbf{u} reads:

$$\mathbf{u} = \left(\frac{\partial \psi}{\partial y}, -\frac{\partial \psi}{\partial x} \right) \quad (7.61)$$

As in [47], the calculation was performed on a 256^2 Cartesian mesh with a constant time step of $\Delta t = 0.002s$. The predicted level set contours are given in Figure 7.5 at $t = \frac{T}{2}$, when the stretching is maximal. As can be seen in Figure 7.5, the RCLS clearly outperforms the ACLS of Desjardins et al. on this test case.

7.6 Interpretation of the method

The RCLS method transports a scalar φ defined as the hyperbolic tangent profile of the signed distance to the interface. The scalar φ is transported in a conservative manner (see Chapter 6) such that it remains conserved to machine accuracy (Lax and Wendroff provided some theoretical background for this assertion in [121] and our numerical experiments, presented in Section 8.4, confirm it). In addition, the transition between phases, characterised by this hyperbolic tangent profile, is kept to a constant thickness.

If applied in the same spirit as Olsson and Kreiss [163] and Desjardins et al. [47], the RCLS method transports a function of the distance to the interface. The phase boundary being defined by the contour $\varphi = 0.5$, all the volume encompassed by that surface can be considered as filled with liquid. Such an interpretation of φ leads to a level set formulation of the RCLS.

However, as the interface thickness is kept constant, the scalar transported φ can be identified to the liquid volume fraction (see Section 7.1). When interpreting φ as such, the RCLS resembles a VOF method with an interface of constant thickness. The interface reconstruction generally required by VOF methods is no longer required here, as the hyperbolic tangent profile provides a smooth field to calculate the interface gradient.

The RCLS method can therefore be perceived as both a level set method and a volume of fluid method.

Chapter 8

A mass-conservative method for efficient atomisation modelling in parallel

At run time, the modelling capability developed involves three main building blocks: the transport of the level set scalar φ , the pressure-velocity coupling and the output of droplet boundary conditions for Lagrangian spray models. All of these operations are performed in a conservative way such that throughout the whole computation no mass is lost.

The transport of the level set is the main item of this work and has already been described in chapters 6 and 7. In this chapter we present the additional building blocks that make the code readily applicable to the atomisation of liquid sheets. In particular, the solution of the incompressible Navier-Stokes equations is detailed in Section 8.1, the algorithm for the droplet transfer is explained in Section 8.2 and the parallelisation of the whole code is reviewed in Section 8.3. Finally, the performance of the modelling capability on typical two-phase flow test cases is reported in Section 8.4.

8.1 Solution of the incompressible Navier-Stokes equations

8.1.1 Conservative formulation

The governing equations for multiphase flows with interface modelling are given in Chapter 3. Implementing in these equations the CSF description of the surface tension (see Chapter 3), leads to the following conservation formulation of the incompressible Navier-Stokes equations:

$$\nabla \cdot \mathbf{u} = 0 \quad (8.1)$$

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla p + \rho\mathbf{g} + \sigma\kappa\nabla\varphi + \nabla \cdot (2\mu\mathcal{D}) \quad (8.2)$$

where the rate-of-strain tensor \mathcal{D} reads:

$$\mathcal{D} = \frac{1}{2} \left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T \right) \quad (8.3)$$

In his implementation of the incompressible Navier-Stokes equations, Weller reformulates some terms of the momentum equation (8.2). In particular, the viscous stress term is re-expressed to improve the efficiency of its numerical determination [208]:

$$\begin{aligned} \nabla \cdot (2\mu\mathcal{D}) &= \nabla \cdot \left(\mu \left(\nabla\mathbf{u} + (\nabla\mathbf{u})^T \right) \right) \\ &= \nabla \cdot (\mu\nabla\mathbf{u}) + (\nabla\mathbf{u}) \cdot \nabla\mu + \mu\nabla(\nabla \cdot \mathbf{u}) \\ &= \nabla \cdot (\mu\nabla\mathbf{u}) + (\nabla\mathbf{u}) \cdot \nabla\mu \end{aligned} \quad (8.4)$$

Similarly, a modified pressure p_d is introduced to simplify the specification of the pressure boundary conditions [208]. p_d is defined by:

$$p_d = p - \rho\mathbf{g} \cdot \mathbf{x} \quad (8.5)$$

The gradient of p_d then reads:

$$\nabla p_d = \nabla p - \rho\mathbf{g} - (\mathbf{g} \cdot \mathbf{x}) \nabla\rho \quad (8.6)$$

Finally, inserting (8.4) and (8.6) into (8.2) leads to the formulation of the momentum equation as used in the code:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = -\nabla p_d + \nabla \cdot (\mu\nabla\mathbf{u}) + (\nabla\mathbf{u}) \cdot \nabla\mu - (\mathbf{g} \cdot \mathbf{x}) \nabla\rho + \sigma\kappa\nabla\varphi \quad (8.7)$$

Before describing the solution procedure for the system of equations formed by (8.1) and (8.7), let us consider the system of linear algebraic equations resulting from the discretisation of these governing equations.

8.1.2 Systems of linear algebraic equations

Although the methodology presented in this sub-section applies directly to any tensorial quantity, for the sake of clarity we will consider a vector variable \mathbf{u} . For each control volume E_i of a mesh that contains N cells, the discretisation procedure of the governing equation for \mathbf{u} leads to the following algebraic equation at the time t_n :

$$a_i\mathbf{u}_i^n + \sum_j a_j\mathbf{u}_j^n = \mathbf{r}_i \quad (8.8)$$

where \mathbf{u}_i and \mathbf{u}_j refer to the value of \mathbf{u} in, respectively, the considered element E_i and its neighbours E_j . The coefficients a_i , a_j and the vector \mathbf{r}_i result from the discretisation of the governing equation for \mathbf{u} .

As the solution in the considered cell \mathbf{u}_i depends on the solution in its neighbouring cells \mathbf{u}_j , writing the equation (8.8) for each control volume produces a systems of algebraic equations. This system can be expressed in the following matrix form:

$$\mathcal{A}\mathbf{U} = \mathbf{R} \quad (8.9)$$

where:

\mathcal{A} is a $N \times N$ matrix of scalars. This square matrix is sparse and is made of the a_i coefficients on the diagonal and of the a_j coefficients elsewhere.

\mathbf{U} is a vector of vectors. This column vector gathers N vectors, one for each cell in the mesh. In particular, the i^{th} row of the vector \mathbf{U} contains \mathbf{u}_i : the solution in the cell E_i .

\mathbf{R} is a vector of vectors. This column vector gathers N vectors, one for each cell in the mesh. Similarly to \mathbf{U} , the i^{th} row of the vector \mathbf{R} contains \mathbf{r}_i : the source vector for the cell E_i .

The matrix \mathcal{A} can be split into two matrices: a diagonal matrix \mathcal{A}_D that gathers the coefficients a_i and the matrix \mathcal{A}_{OD} which contains the off-diagonal coefficients a_j :

$$\mathcal{A} = \mathcal{A}_D + \mathcal{A}_{OD} \quad (8.10)$$

\mathcal{A}_D being diagonal, it only contains N scalar elements that can be stored in the column vector \mathbf{A}_D .

Let us now introduce the vector of vectors \mathbf{A}_H , defined by:

$$\mathbf{A}_H = \mathbf{R} - \mathcal{A}_{OD}\mathbf{U} \quad (8.11)$$

The next sub-section, which describes the handling of the pressure-velocity coupling, involves semi-implicit discretisations of the equations (8.1) and (8.7). In that solution procedure, the data contained in \mathbf{A}_D and \mathbf{A}_H is used in conjunction with terms of the momentum equation treated explicitly. In these data manipulations, \mathbf{A}_D and \mathbf{A}_H should no longer be perceived as vectors but as fields: \mathbf{A}_D corresponding to a scalar field and \mathbf{A}_H to a vector field. In order to remain coherent in terms of notation, we note the scalar field relating to \mathbf{A}_D : a_D and the vector field relating to \mathbf{A}_H : \mathbf{a}_H .

8.1.3 Pressure-velocity coupling

The incompressible Navier-Stokes equations are discretised in a semi-implicit manner such that the restriction on the time step remains low. The pressure velocity coupling is handled with the pressure-implicit with splitting operators (PISO)

method of Issa [99]. This non-iterative method proceeds through a series of predictor and corrector steps that approximate the exact velocity and pressure fields with improving accuracy as the number of PISO loops increases.

The method as implemented in the code involve three steps: the “momentum predictor”, the “pressure solution” and the “explicit velocity correction” [103]. The last two steps being iterated as many times as required to reach a given accuracy. Issa demonstrates that two loops are sufficient if the equations are discretised with second-order accurate schemes [99].

In his implementation of the PISO algorithm, Weller uses the principle of the pseudo-staggered grid of the Rhie and Chow interpolation [199] to remove the checker-board pressure oscillation. In that procedure, Weller introduces the volumetric flow rate Q stored at the face centres. The scalar Q_l associated to the face F_l corresponds to:

$$Q_l = \mathbf{u}_l \cdot \mathbf{F}_l \quad (8.12)$$

where $\mathbf{F}_l = |F_l| \mathbf{n}_l$

However, Q_l is not calculated directly from (8.12) because the velocity field \mathbf{u} being a collocated variable, \mathbf{u}_l would have to be interpolated and the field Q would not satisfy the continuity equation.

It is convenient to use the volumetric flow rate as a variable in a finite volume framework for incompressible flows because this quantity often appears in the discretised formulations of the equations through the application of Gauss’ theorem.

Momentum predictor

In order to calculate the predicted velocity Weller applies the following semi-implicit discretisation to (8.7):

$$\left[\frac{\partial \rho [\mathbf{u}]}{\partial t} \right] + \llbracket \nabla \cdot (\rho \mathbf{u} \otimes [\mathbf{u}]) \rrbracket = -\nabla p_d + \llbracket \nabla \cdot (\mu \nabla [\mathbf{u}]) \rrbracket + (\nabla \mathbf{u}) \cdot \nabla \mu - (\mathbf{g} \cdot \mathbf{x}) \nabla \rho + \sigma \kappa \nabla \varphi \quad (8.13)$$

In (8.13), we are using same convention as Weller's [208, 272], which notes in double brackets the terms treated implicitly and in single brackets the sought variable of the equation. Therefore in (8.13), while the unsteady term, the convection term and the diffusion term are treated implicitly, all the other terms are treated explicitly. Besides, only the velocity vectors in single brackets, $[\mathbf{u}]$, are unknown. As all other quantities are taken from the previous time step, the notation implies that the convection term is linearised.

Let us detail how the convection and the diffusion terms are discretised:

- The volume integration and the linearisation of the convection term are applied the following way:

$$\begin{aligned}
 \iiint_{E_i} \nabla \cdot (\rho \mathbf{u} \otimes [\mathbf{u}]) \, dE_i &= \sum_{l=1}^{L_i} \mathbf{F}_l \cdot (\rho \mathbf{u} \otimes [\mathbf{u}])_l \\
 &= \sum_{l=1}^{L_i} (\mathbf{F}_l \cdot \mathbf{u}_l) (\rho [\mathbf{u}])_l \\
 &= \sum_{l=1}^{L_i} Q_l (\rho [\mathbf{u}])_l
 \end{aligned} \tag{8.14}$$

in which Q_l , taken from the previous time step, satisfies the continuity equation.

- The volume integration of the diffusion term leads to:

$$\begin{aligned}
 \iiint_{E_i} \nabla \cdot (\mu \nabla [\mathbf{u}]) \, dE_i &= \sum_{l=1}^{L_i} \mathbf{F}_l \cdot (\mu \nabla [\mathbf{u}])_l \\
 &= \sum_{l=1}^{L_i} \mu_l \mathbf{F}_l \cdot (\nabla [\mathbf{u}])_l
 \end{aligned} \tag{8.15}$$

NB: All throughout this section, the subscript l refers to the face value of the associated quantity.

The solution of the discretised equation (8.13) produces a first guess for the velocity: \mathbf{u}^* (as noted by Issa in [99]) which does not satisfy the continuity equation.

Pressure solution

Let us write below the momentum equation without the surface tension force, the gravity force and the pressure gradient:

$$\frac{\partial(\rho\mathbf{u})}{\partial t} + \nabla \cdot (\rho\mathbf{u} \otimes \mathbf{u}) = \nabla \cdot (\mu\nabla\mathbf{u}) + (\nabla\mathbf{u}) \cdot \nabla\mu \quad (8.16)$$

The equation (8.16) is discretised in the same semi-implicit way as was the equation (8.7) in the momentum predictor step:

$$\left[\left[\frac{\partial\rho[\mathbf{u}]}{\partial t} \right] \right] + \llbracket \nabla \cdot (\rho\mathbf{u} \otimes [\mathbf{u}]) \rrbracket = \llbracket \nabla \cdot (\mu\nabla[\mathbf{u}]) \rrbracket + (\nabla\mathbf{u}) \cdot \nabla\mu \quad (8.17)$$

As described in Section 8.1.2, a semi-implicitly discretised equation such as (8.17) results in a system of algebraic equations that can be put in a matrix form. We note \mathcal{A} , the matrix of that algebraic system.

Two fields are then constructed from the system of algebraic equations resulting from (8.17) (see Section 8.1.2):

- A scalar field a_D gathering the diagonal coefficients of \mathcal{A} .
- A vector field \mathbf{a}_H built from the off-diagonal coefficients of \mathcal{A} , the source vector and the predicted velocity field \mathbf{u}^* .

Introducing a_D and \mathbf{a}_H in (8.13), leads to the following semi-discretised formulation of the momentum equation:

$$\mathbf{u} = \frac{1}{a_D} (\mathbf{a}_H - (\mathbf{g} \cdot \mathbf{x}) \nabla\rho + \sigma\kappa\nabla\varphi) - \frac{1}{a_D} \nabla p_d \quad (8.18)$$

The first term on the right hand side of (8.18) is homogeneous to a velocity, but lacks the contribution of the pressure gradient to satisfy the momentum equation. We denote it as \mathbf{u}^{NP} and rewrite (8.18) as:

$$\mathbf{u} = \mathbf{u}^{NP} - \frac{1}{a_D} \nabla p_d \quad (8.19)$$

Expressing the velocities on the cell face \mathbf{u}_l as the face interpolation of (8.19) leads to:

$$\mathbf{u}_l = \mathbf{u}_l^{NP} - \left(\frac{1}{a_D} \right)_l (\nabla p_d)_l \quad (8.20)$$

As \mathbf{u} is now sought such that it satisfies the continuity equation, introducing (8.19) in (8.1) leads to:

$$\nabla \cdot \mathbf{u}^{NP} = \nabla \cdot \left(\frac{1}{a_D} \nabla p_d \right) \quad (8.21)$$

Calculating the volume integration of (8.21) using Gauss' theorem results in:

$$\sum_{l=1}^{L_i} \mathbf{F}_l \cdot \mathbf{u}_l^{NP} = \sum_{l=1}^{L_i} \mathbf{F}_l \cdot \left(\left(\frac{1}{a_D} \right)_l (\nabla p_d)_l \right) \quad (8.22)$$

$$\sum_{l=1}^{L_i} Q_l^{NP} = \sum_{l=1}^{L_i} \mathbf{F}_l \cdot \left(\left(\frac{1}{a_D} \right)_l (\nabla p_d)_l \right) \quad (8.23)$$

where Q_l^{NP} is the volumetric flow rate associated to \mathbf{u}_l^{NP} . Weller calculates Q_l^{NP} according to:

$$Q_l^{NP} = \left(\frac{1}{a_D} \right)_l \left(\mathbf{F}_l \cdot (\mathbf{a}_H)_l - (\mathbf{g} \cdot \mathbf{x})_l \mathbf{F}_l \cdot (\nabla \rho)_l + (\sigma \kappa)_l (\nabla \varphi)_l \right) \quad (8.24)$$

The solution of (8.23) provides the new pressure field noted p_d^* by Issa in [99].

Explicit velocity correction

The volumetric flow rates are then updated according to:

$$\begin{aligned} Q_l &= \mathbf{F}_l \cdot \mathbf{u}_l \\ &= \mathbf{F}_l \cdot \left(\mathbf{u}_l^{NP} - \left(\frac{1}{a_D} \right)_l (\nabla p_d)_l \right) \\ &= Q_l^{NP} - \mathbf{F}_l \cdot \left(\left(\frac{1}{a_D} \right)_l (\nabla p_d)_l \right) \end{aligned} \quad (8.25)$$

As (8.22) is satisfied, the updated volumetric flow rates are conservative.

Weller then reconstructs the velocity field from Q starting with the following assumption:

$$\sum_{l=1}^{L_i} \mathbf{n}_l (\mathbf{F}_l \cdot \mathbf{u}) = \sum_{l=1}^{L_i} \mathbf{n}_l (\mathbf{F}_l \cdot \mathbf{u}_l) \quad (8.26)$$

Now, let us introduce the symmetric tensor $\mathcal{D}_l = \mathbf{n}_l \otimes \mathbf{S}_l$. The dyad \mathcal{D}_l has the following property:

$$\begin{aligned} \mathcal{D}_l \cdot \mathbf{u} &= (\mathbf{n}_l \otimes \mathbf{F}_l) \cdot \mathbf{u} \\ &= \mathbf{n}_l (\mathbf{F}_l \cdot \mathbf{u}) \end{aligned} \quad (8.27)$$

where \mathcal{D}_l maps the velocity vector \mathbf{u} onto a vector parallel to \mathbf{n}_l with a magnitude $|\mathbf{n}_l| (\mathbf{F}_l \cdot \mathbf{u}) \approx Q_l$. Therefore \mathcal{D}_l maps \mathbf{u} onto a vector $\approx \mathbf{n}_l Q_l$.

Introducing (8.27) into (8.26), leads to:

$$\sum_{l=1}^{L_i} \left(\underbrace{(\mathbf{n}_l \otimes \mathbf{F}_l)}_{\mathcal{D}_l} \cdot \mathbf{u} \right) = \sum_{l=1}^{L_i} \mathbf{n}_l (\mathbf{F}_l \cdot \mathbf{u}_l) \quad (8.28)$$

Applying to the left hand side of (8.28) the distributivity of the inner product of a tensor by a vector, and replacing $\mathbf{F}_l \cdot \mathbf{u}_l$ by the volumetric flow rate Q_l on the right hand side of (8.28) leads to:

$$\left(\sum_{l=1}^{L_i} \mathcal{D}_l \right) \cdot \mathbf{u} = \sum_{l=1}^{L_i} \mathbf{n}_l Q_l \quad (8.29)$$

Let us now introduce the symmetric tensor \mathcal{D} defined by:

$$\mathcal{D} = \sum_{l=1}^{L_i} \mathcal{D}_l \quad (8.30)$$

Since \mathcal{D} is symmetric and made of real coefficients, it can be inverted. Multiplying on either side of (8.29) by \mathcal{D}^{-1} finally provides the reconstructed velocity field:

$$\begin{aligned} \mathbf{u} &= \mathcal{D}^{-1} \cdot \left(\sum_{l=1}^{L_i} \mathbf{n}_l Q_l \right) \\ &= \left(\sum_{l=1}^{L_i} \mathbf{n}_l \otimes \mathbf{F}_l \right)^{-1} \cdot \left(\sum_{l=1}^{L_i} \mathbf{n}_l Q_l \right) \end{aligned} \quad (8.31)$$

The velocity field produced by equation (8.31) is the new velocity field denoted as \mathbf{u}^{**} by Issa in [99]. The solution procedure then repeats the pressure solution and the explicit velocity correction steps a given number of times. Each new iteration of the PISO loop starts from the pressure and velocity fields calculated at the previous PISO iteration.

Typically, the systems of algebraic equations produced for the pressure-velocity coupling are solved by using either the pre-conditioned conjugate gradient (PCG) or the generalised geometric-algebraic multi-grid (GAMG).

8.2 Droplet transfer

8.2.1 Motivations

The need for droplet transfer arises from the computational modelling of atomisation. As described in Chapter 2, the process of atomisation transforms a jet of liquid into droplets. It generally involves two stages: the primary breakup where drops are detaching from the core of the liquid jet and the secondary breakup where the droplets produced at the previous stage further disintegrate into smaller droplets.

Although there are some reliable secondary breakup models, the primary breakup — highly dependent on the geometry of the injection device — has to be modelled directly using interface description methods like RCLS. However, the faithful modelling of the spray requires the simulation of both the primary and the secondary breakups. In order to reduce the computational cost of the modelling capability Kim et al. [111] had the idea to remove the droplets from the level set formulation to send them into a Lagrange tracking framework equipped with an appropriate secondary breakup model.

This strategy allows the modelling capability to represent both types of breakup in the same computation while significantly reducing the computational cost. Indeed, as illustrated in Section 4.3, under aero-engine conditions, the droplet size can reduce

to very small values and the mesh refinement required to resolve such small liquid structures is unaffordable.

8.2.2 Outline of the method

In our implementation, the droplets satisfying certain criteria (see Section 8.2.4) are removed from the level set field φ at specific times t_R^n . In order to reduce the computational cost, this transfer only occurs at the output time step (time step at which the fields computed are written out by the code). This strategy aims at providing a better trade-off in terms of computational cost versus resolution of droplets.

Once identified and removed from the field φ , the droplets are then stored in two ways:

- The level set field associated with the removed droplets is stored into a “Drops” field. Therefore, “Drops” is the level set field gathering all the drops produced between t_R^{n-1} and t_R^n in the state in which they were at t_R^n . The field “Drops” has two purposes: it allows us to conduct further analysis on the droplets (calculation of specific geometrical parameters, etc...) and gives the option to restart from t_R^n with the level set field as it was before the droplet removal.
- Some characteristics of the droplets removed at t_R^n are calculated and stored in lists (Section 8.2.5). These data are the boundary conditions required by Lagrange tracking schemes to simulate the spray with a secondary breakup model. Once produced on a sub-model of a fuel injector, these lists can be imported by combustion codes for the simulation of the reacting flow in a combustion chamber.

The algorithm operates the transfer of droplets in five steps:

1. Blob search: this routine searches through the computational domain to identify coherent liquid structures of any shape and size.

2. Drop selection: the algorithm searches through the blobs identified and selects the ones that match the droplet criteria (see Section 8.2.4). Because the selection criteria are based on droplet characteristics, the generation of the boundary conditions for the Lagrange tracking schemes is also performed during this step.
3. Transfer drops: the level set field associated to the droplets is removed from the field φ . After this step, the hyperbolic tangent profile is lost.
4. Write drop characteristics: the boundary conditions for the combustion codes are written out.
5. Re-initialisation of φ : the level set field is re-initialised in order to reconstruct the hyperbolic tangent profile of φ . As the re-initialisation is done in a conservative way, this step does not affect the mass conservation properties of the modelling capability.

8.2.3 Identification of blobs

Definition of a blob

We consider blobs as liquid structures of arbitrary shapes, sizes, positions and velocities. The search algorithm defines a blob \mathcal{B} as a group of cells E_i that satisfies the following criteria (see Figure 8.1 for a 1D representation of the criteria given below):

- The cells E_i are face-neighbours, i.e. each cell of the blob shares at least one face with another cell of the blob (see Figure 8.3 for an illustration on a 2D mesh.).
- In each cell of the blob, the liquid volume fraction is above a given threshold φ_{Lmin} , such that it can be considered as containing some liquid. This condition

reads:

$$\forall E_i \in \mathcal{B} : \quad \varphi_i \geq \varphi_{Lmin} \quad (8.32)$$

- All the cells E_k surrounding the blob \mathcal{B} — i.e. sharing at least one face with a cell E_i of \mathcal{B} — have a liquid volume fraction below φ_{Lmin} , such that they can be considered as “gas”. This condition reads:

$$\forall E_k \in \{\text{face-neighbours of } E_i\} : \quad \varphi_k < \varphi_{Lmin} \quad (8.33)$$

Search algorithm

The search algorithm starts by storing all the cells containing liquid into the list “liqRest”. Then, as long as the list “liqRest” is not empty, the routine performs the following operations:

1. Consider the first element of “liqRest”: E_1
2. Add layers of cells around E_1 in a similar way that cells were gathered for the generation of a central stencil in Chapter 6 (see Section 6.3.1). However, here the layers consist of face-neighbours of the cells belonging to the previous layer, that are not already in the blob and that contain some liquid (see condition (8.32)). The addition of layers is continued until the algorithm produces an empty list for the current layer. At that point, no face-neighbour with a liquid volume fraction greater or equal to φ_{Lmin} could be found by the algorithm, such that the blob is surrounded by “gas”.
3. Remove the cells of the blob constituted from the list “liqRest”. As a result, that list only gathers cells that contain liquid and that have not yet been attributed to a blob.

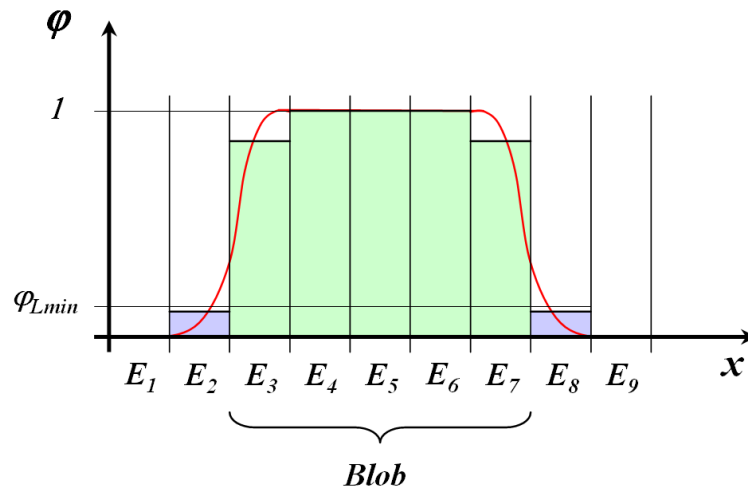


Figure 8.1: Schematic of a blob on a 1D mesh — Hyperbolic tangent profile (in red), liquid volume fraction in the blob (light green) and liquid volume fraction outside of the blob (in purple)

8.2.4 Selection of drops

Definition of a droplet

Once the list of blobs constructed from the level set field φ , the selection algorithm identifies the liquid structures that qualify as droplets. We choose to define a droplet \mathcal{D} as a group of $N_{\mathcal{D}}$ cells E_i that satisfies the following criteria:

- A droplet contains at least one cell with a volume fraction above a given threshold: $\varphi_{\mathcal{D}min}$.
- A droplet has an equivalent diameter D_{eq} smaller than a given threshold: D_{max} .
- A droplet has to be fairly spherical.
- As it is about to be sent into a Lagrange tracking scheme, a droplet should not be on the verge of coalescing with the core of the liquid jet \mathcal{C} (largest liquid structure).

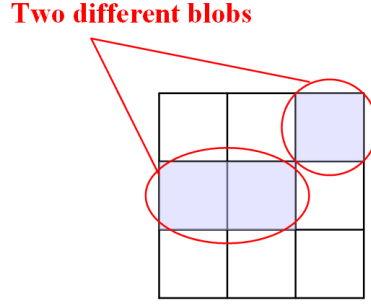


Figure 8.2: Illustration of the principle of face-neighbours used to define blobs

The first criterion can be expressed the following way:

$$\exists E_j \in \mathcal{D} \mid \varphi_j \geq \varphi_{\mathcal{D}min} \quad (8.34)$$

As mentioned in Section 7.2, the solution of the advection equation diffuses the hyperbolic tangent profile, such that patches of cells containing small amounts of liquid may appear in the solution. Although the re-initialisation of the profile will get rid of these patches, the user has the option to solve the re-initialisation equation only periodically and not at every time step (see Section 9.2.2). In the case of a droplet transfer occurring between two re-initialisation steps, the criterion (8.34) guarantees that these patches are not mistaken for droplets.

The limitation on the droplet size (second criterion) reads:

$$D_{eq} \leq D_{max} \quad (8.35)$$

Introducing the volume of the droplet $V_{\mathcal{D}}$, the equivalent diameter reads:

$$D_{eq} = \sqrt[3]{\frac{6V_{\mathcal{D}}}{\pi}} \quad (8.36)$$

This maximum diameter can be set by the user either as a multiple of Δx (characteristic length scale of the mesh) or as an actual dimension in metres. In order to simulate directly the secondary breakup or to account for the coalescence of the blobs, the user may want to keep in the level set formulation all the liquid structures

that can be resolved by the mesh. However, depending on the parameters of the Lagrangian spray model and on the targeted computational cost, it may be necessary to increase D_{max} .

After Kim et al. [111], we choose the following expression for the sphericity criterion:

$$r_{max} < D_{eq} \quad (8.37)$$

The condition (8.37) formulates that the maximum radius of the droplet r_{max} is smaller than the droplet's equivalent diameter. We define r_{max} as the maximum distance between the centre of mass of the droplet $C_{\mathcal{D}}$ and the centre of a cell belonging to the droplet C_i . The expression for r_{max} reads:

$$r_{max} = \max_{E_i \in \mathcal{D}} \left(\left| \overrightarrow{C_{\mathcal{D}}C_i} \right| \right) \quad (8.38)$$

This criterion ensures that non-spherical blobs — such as detached ligaments — remain in the level set field. Typically, detached ligaments would undergo Plateau-Rayleigh instabilities and breakup into a row of droplets. Such breakup phenomena are not generally accounted for by Lagrangian spray models.

Berlemont has observed small broken-off droplets flowing back into the core of the liquid jet in DNS studies of round jet atomisation [1]. To account for this phenomenon we have introduced a direction criterion in the droplet selection (see Figure 8.3 for an illustration in 2D). Introducing the droplet and liquid core centres of mass, respectively $C_{\mathcal{D}}$ and C_c , and the relative droplet velocity (with respect to the liquid core): $\overrightarrow{\Delta u} = \mathbf{u}_{\mathcal{D}} - \mathbf{u}_c$, we express the direction criterion the following way:

$$\overrightarrow{\Delta u} \cdot \overrightarrow{C_c C_{\mathcal{D}}} > 0 \quad (8.39)$$

In order to account for both the droplets flowing towards the liquid core and the droplets being caught up by the liquid jet, the criterion involves the dot product of the relative droplet velocity with the relative droplet position (with respect to the centre of mass of the liquid core). Although this criterion does not guarantee to

leave in the level set field all the droplets that would coalesce with the liquid core, it is calculated quickly and remains quite effective.

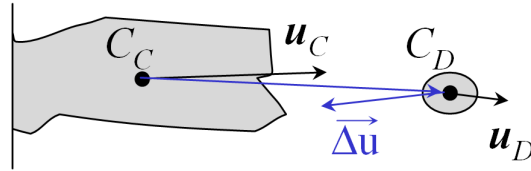


Figure 8.3: Illustration of the direction criterion for the droplet selection — For the pictured velocities and positions of the droplet \mathcal{D} and the liquid core \mathcal{C} , the direction criterion is not met and \mathcal{D} is left in the level set field

Selection algorithm

The selection algorithm progresses through the list of blobs and assesses whether the current blob qualifies as a droplet. To improve the efficiency of the selection process, the routine evaluates the droplet criteria in sequence, performing the most discriminating tests first. As soon as a test is failed, the code skips to the next blob in the list.

First, the routine identifies the core of the liquid jet and computes its properties for the determination of the direction criterion. The liquid core is defined as the largest blob that contains at least N_C cells. The minimum number of cells in the liquid core N_C is either set directly by the user or through the specification of the characteristic length scale of the liquid jet.

Then, the code goes through the list of blobs produced by the search algorithm and performs the following steps:

1. Test whether the current blob \mathcal{B} contains more than N_C elements. If so go to the next blob.
2. Check that the blob contains at least one cell E_j with $\varphi_j \geq \varphi_{Dmin}$. If it does not go to the next blob.

3. Calculate the characteristics of the blobs: volume $V_{\mathcal{B}}$, centre of mass $C_{\mathcal{B}}$, mass-averaged velocity $\mathbf{u}_{\mathcal{B}}$, mass $m_{\mathcal{B}}$ and momentum $\mathbf{p}_{\mathcal{B}}$. As the calculation of these quantities involves going through all the cells in the current blob, they are all performed in the same step.
4. Calculate the equivalent diameter of the blob: $D_{eq\mathcal{B}}$ from $V_{\mathcal{B}}$ using (8.36). Test if $D_{eq\mathcal{B}}$ is larger than the maximum diameter D_{max} . If so, go to the next blob.
5. Calculate the maximum radius of the liquid structure from the blob's centre of mass $C_{\mathcal{B}}$ by applying (8.38). Test the sphericity of the current blob. If it does not satisfy the condition (8.37), go to the next blob.
6. If the domain contains cyclic or processor boundary conditions BC_{cyclic} (see Section 8.3), test if the current blob contains a cell on the border BC_{cyclic} . If so, go to the next blob.
7. Test the direction criterion for the current blob. If it does not satisfy (8.39), go to the next blob.
8. At this point of the loop, the current blob has satisfied all the droplet criteria. Therefore its characteristics, calculated in the previous steps, are stored in the relevant droplet lists.

8.2.5 Drop characteristics of interest

Droplet removal techniques

As mentioned in Section 7.1, the conservative level set method transports a hyperbolic tangent profile φ of the signed distance function from the interface ϕ . As a result, the level set field φ only tends to zero or one, at an infinite distance in, respectively, the gas phase or the liquid phase. We handle this ambiguity in the definition of blobs (Section 8.2.3) by considering that a cell E_i contains some liquid

if the associated level set scalar φ_i is above a minimum threshold φ_{Lmin} . It results, that several options are available to remove the droplets from the level set field.

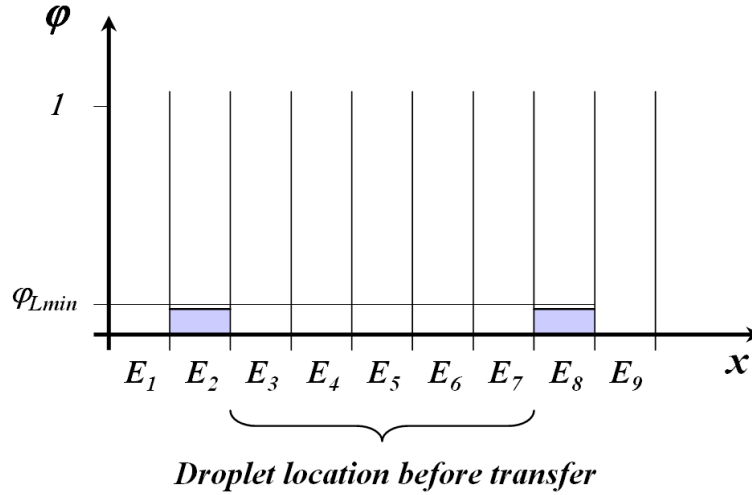


Figure 8.4: Illustration of the first droplet removal technique: “option 1”

The code offers to the user two means of removing the droplets. The first option (“option 1”) is to remove all of the “liquid volume fraction” by setting $\varphi_i = 0$ in all the cells E_i that belong to the considered droplet \mathcal{D} . The other option (“option 2”) is to truncate the profile by removing only $\varphi_i - \varphi_{Lmin}$ from each droplet cell, thus setting $\varphi_i = \varphi_{Lmin}$ in each cell of \mathcal{D} . Figures 8.4 and 8.5 present the fields obtained after removing the blob pictured in Figure 8.1 using respectively the first and the second technique. After the droplet removal the level set field is re-initialised (in a conservative manner), such that the hyperbolic tangent profile is recovered.

Depending on the technique chosen, the droplet characteristics are not calculated in the same way. Indeed, it is crucial to account for liquid volume fraction actually removed in order to maintain mass conservation all the way through the calculation.

Characteristics outputted

Some droplet characteristics are calculated from the amount of liquid in each cell of the droplet. As a result, depending on the option chosen by the user, two alternative

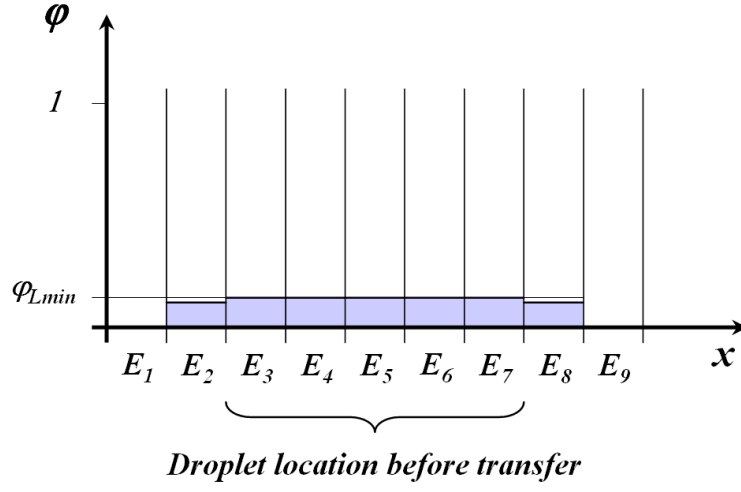


Figure 8.5: Illustration of the second droplet removal technique: “option 2”

formulae exist to compute these quantities.

The droplets characteristics calculated and stored by the algorithm are:

- The volume $V_{\mathcal{D}}$, given by:

$$\text{Option 1:} \quad V_{\mathcal{D}} = \sum_{i=1}^{N_{\mathcal{D}}} |E_i| \varphi_i \quad (8.40)$$

$$\text{Option 2:} \quad V_{\mathcal{D}} = \sum_{i=1}^{N_{\mathcal{D}}} |E_i| (\varphi_i - \varphi_{Lmin}) \quad (8.41)$$

- The mass $m_{\mathcal{D}}$, calculated from $V_{\mathcal{D}}$ according to:

$$m_{\mathcal{D}} = \rho_{liq} V_{\mathcal{D}} \quad (8.42)$$

- The centre of mass $C_{\mathcal{D}}$, expressed as:

$$\text{Option 1:} \quad \overrightarrow{OC_{\mathcal{D}}} = \frac{1}{V_{\mathcal{D}}} \sum_{i=1}^{N_{\mathcal{D}}} |E_i| \varphi_i \overrightarrow{OC_i} \quad (8.43)$$

$$\text{Option 2:} \quad \overrightarrow{OC_{\mathcal{D}}} = \frac{1}{V_{\mathcal{D}}} \sum_{i=1}^{N_{\mathcal{D}}} |E_i| (\varphi_i - \varphi_{Lmin}) \overrightarrow{OC_i} \quad (8.44)$$

- The mass-averaged velocity vector \mathbf{u}_D , given by (8.45) and (8.46) below:

$$\text{Option 1:} \quad \mathbf{u}_D = \frac{1}{V_D} \sum_{i=1}^{N_D} |E_i| \varphi_i \mathbf{u}_i \quad (8.45)$$

$$\text{Option 2:} \quad \mathbf{u}_D = \frac{1}{V_D} \sum_{i=1}^{N_D} |E_i| (\varphi_i - \varphi_{Lmin}) \mathbf{u}_i \quad (8.46)$$

- The mass-averaged momentum vector \mathbf{p}_D , calculated from the mass and the mass-averaged velocity vector using (8.47) below:

$$\mathbf{p}_D = m_D \mathbf{u}_D \quad (8.47)$$

- The equivalent diameter D_{eqD} , calculated according to (8.36).

8.2.6 Test cases

In this section, the performance of the droplet transfer is demonstrated on a static case and on the Rayleigh-Taylor instability.

Illustration of the algorithm

This basic test case consists in the sole application of the droplet transfer on a given level set field populated with blobs of various shape, size, position and of different liquid volume fractions. In this test, no equations are solved such that the only field modified is φ and this modification is done by running the algorithm described in the above sub-sections. It follows that the direction criterion cannot be tested by this case.

This test is performed on a the domain $D = [-0.5; 0.5] \times [-2; 2]$ metres — in the frame of reference $(O, \mathbf{x}, \mathbf{y})$ — meshed with 9234 triangles and run both in serial and in parallel. For the parallel run, the domain is decomposed in four sub-domains by cutting halfway along the \mathbf{x} and \mathbf{y} directions. The settings of the droplet transfer for both the serial and parallel runs are given in Table 8.1.

Table 8.1: Settings of the droplet transfer algorithm for the static test case

| | |
|---------------------------|----------------------------------|
| Liquid threshold: | $\varphi_{Lmin} = 0.01$ |
| Droplet threshold: | $\varphi_{\mathcal{D}min} = 0.5$ |
| Liquid core size (cells): | $N_{\mathcal{C}} = 1000$ |
| Maximum diameter (m): | $D_{max} = 0.2$ |
| Removal technique: | Option 2 |

For the serial case, the level set field is initialised with the five following blobs (from top to bottom in Figure 8.6):

- A small, circular blob with a liquid volume fraction $\varphi_{\mathcal{B}} = 1$. This blob is the only one that qualifies as a droplet.
- A thin ligament with a liquid volume fraction $\varphi_{\mathcal{B}} = 0.6$.
- A very large circular blob with a liquid fraction $\varphi_{\mathcal{B}} = 1$. Because of its size, this liquid structure is recognised as the core of the liquid jet by the algorithm.
- A large circular blob with a diameter larger D_{max} .
- A small circular blob with a low liquid volume fraction: $\varphi_{\mathcal{B}} = 0.4$.

The initial field described above allows us to illustrate all the tests performed by the selection algorithm except the direction and the cyclic boundary tests. As expected, the algorithm only removes the small circular blob with $\varphi_{\mathcal{B}} = 1$, located in the top of the domain.

As the five blobs of the serial case sit on the processor boundaries, four blobs are added in the level set field of the parallel case. These blobs are each placed well inside a different sub-domain. These additional liquid structures are copies of the

initial blobs: the droplet, the thin ligament, the large circular blob and the blob with $\varphi_B = 0.4$ (see Figure 8.6).

As expected, the droplet located on the processor boundary fails the cyclic boundary test and only the droplet well inside its sub-domain is removed from the level set field.

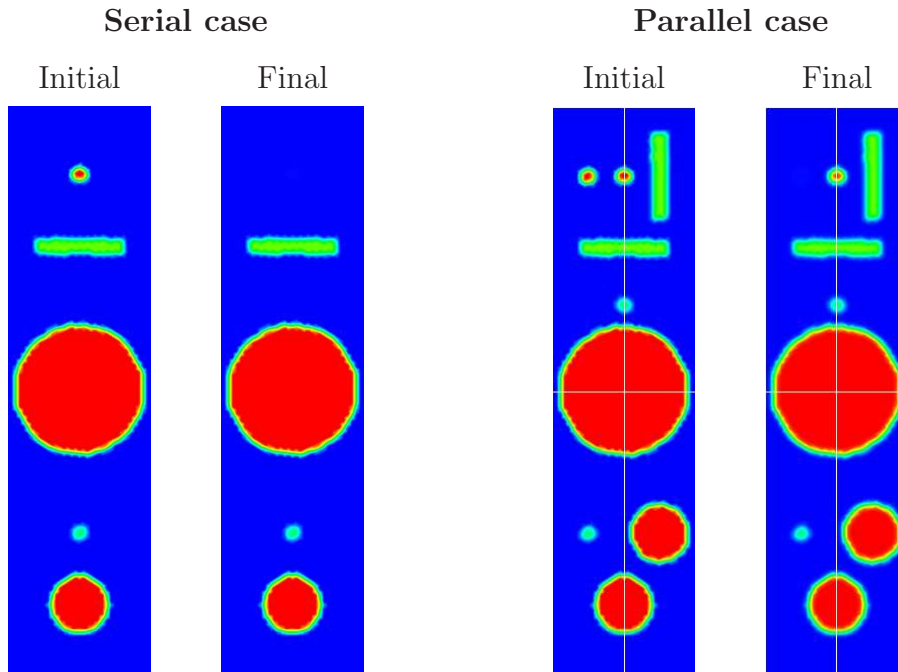


Figure 8.6: Illustration of the droplet transfer on the static test case — On the left: serial case; on the right: parallel case (sub-domain boundaries drawn in white)

Rayleigh-Taylor instability

This two-phase flow problem is performed on the same triangular mesh as the static case presented above. The physical parameters for the Rayleigh-Taylor instability are given in Section 4.2.2 and the settings for the droplet transfer are given in Table 8.2 below. As the Rayleigh-Taylor instability case only produces droplets flowing towards the main liquid blob, the direction criterion is toggled off in this simulation.

Table 8.2: Settings of the droplet transfer algorithm for the Rayleigh-Taylor instability case

| | |
|----------------------------------|------------------------|
| Liquid threshold: | $\varphi_{Lmin} = 0.1$ |
| Droplet threshold: | $\varphi_{Dmin} = 0.5$ |
| Liquid core size (cells): | $N_C = 1000$ |
| Maximum diameter (Δx): | $D_{max} = 20$ |
| Removal technique: | Option 1 |

The level set fields produced by the computation at time $t = 1.2s$ — φ before and after the droplet transfer and “Drops” — are given in Figure 8.7. As illustrated, the algorithm only removes from φ the liquid structures qualifying as droplets.

8.3 Outline of the parallel implementation

A significant amount of effort has been put in to make the whole modelling capability parallelisable. Indeed, while the code comes with a 0-halo parallelisation method, the correct operation of the high-order WENO scheme (see Chapter 6) requires a n-halo approach.

0-halo parallelisation of OpenFOAM

The 0-halo approach implemented in OpenFOAM consists in dividing the global domain D in non-overlapping sub-domains D_i (see Figure 8.8). The transfer of information from one sub-domain to the other is therefore performed through the sub-domain boundaries only. It indicates that the numerical schemes used to discretise the governing equations become at best second-order accurate at the inter-processor boundaries.

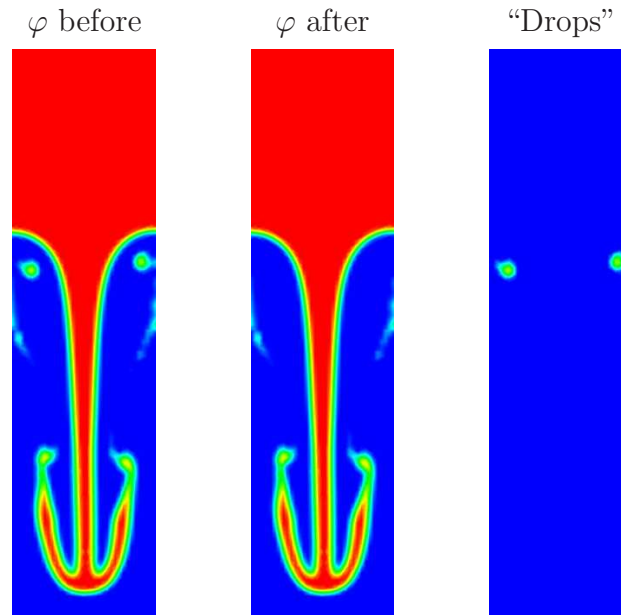


Figure 8.7: Illustration of the droplet transfer on the Rayleigh-Taylor instability case — On the left: level field before droplet removal; in the middle: level set field after droplet removal; on the right: “Drops” field

Although this methodology is appropriate for the solution of the incompressible Navier-Stokes equations discretised with second-order schemes, the 0-halo parallel transport of the level set, using a high-order scheme, leads to non-physical behaviour of the interface at the inter-processor boundaries.

n-halo parallel transport of the level set

The n-halo parallelisation of the RCLS alone was first performed by simply producing a n-halo decomposition of the domain (see Figure 8.9) and updating in each extended sub-domain Dh_i , the halo cells according to the solution on the neighbouring sub-domains Dh_j . The pre-processing variables are then produced for the extended sub-domains such that the WENO scheme performs with the sought high-order accuracy on the non-extended part of Dh_i .

The halos are produced by gathering layers of cells around the non-extended sub-

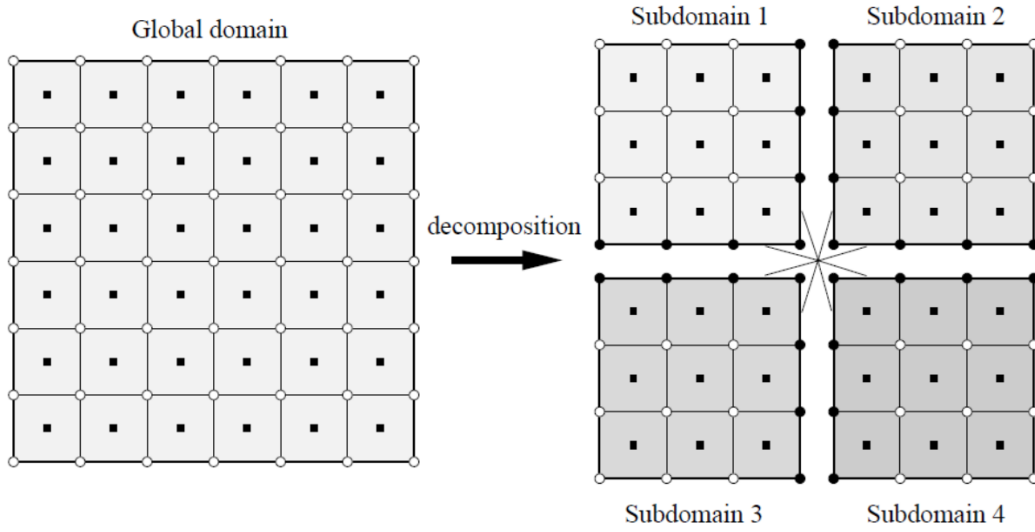


Figure 8.8: Decomposition of the domain using a 0-halo approach [104] — On the left the global domain; on the right the 0-halo sub-domains

domain D_i . The number of layers N_l is a function of the order of the scheme and is calculated such that the solution in the first layer of halo cells L_1 , is high-order accurate. This ensures that the flux exchanged between D_i and L_1 is high-order accurate, such that the solution is properly calculated in all the cells of the non-extended part of Dh_i .

However, the modelling capability involves not only the transport of the level set but also the solution of the pressure-velocity coupling. As the incompressible Navier-Stokes equations are discretised semi-implicitly (see Section 8.1.3), the resulting system of algebraic equations (see Section 8.1.2) is solved by linear solvers. These linear solvers are implemented in OpenFOAM for a 0-halo parallelisation and can only run on a domain decomposed into non-overlapping domains.

n-halo parallel simulation of two-phase flows

An extension of the above n-halo methodology has been produced to transport the level set with high-order accuracy and solve, in the same computation, the in-

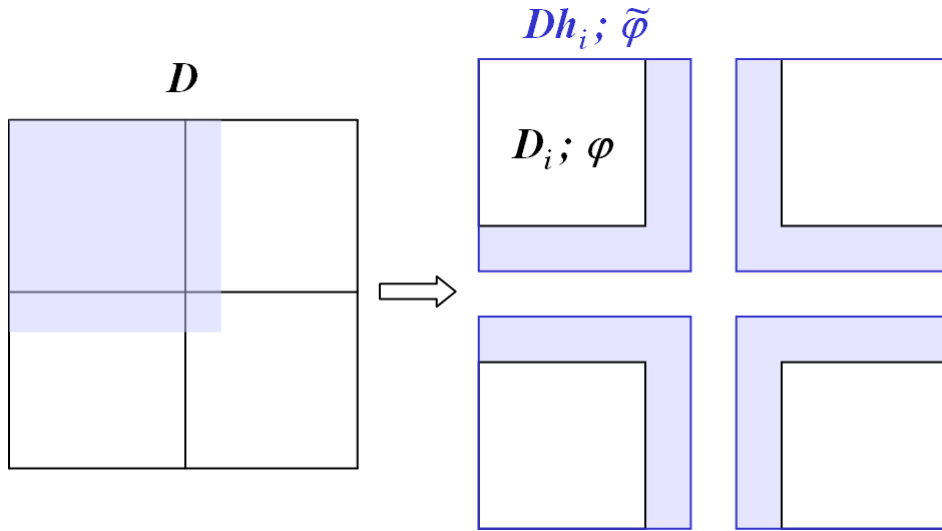


Figure 8.9: Decomposition of the domain using a n-halo approach — On the left the global domain (0-halo inter-processor boundaries in black; extent of a n-halo sub-domain in grey); on the right the 0-halo sub-domains D_i (black line), the extended sub-domains Dh_i (blue line) and the halos (grey region)

compressible Navier-Stokes equations with the linear solvers implemented in OpenFOAM.

Because of these linear solvers, the calculation has to run on a domain decomposed with a 0-halo approach. In order to maintain the high-order accuracy of the level set transport, the pre-processing variables are calculated on the extended sub-domains and re-written for the 0-halo sub-domains. This re-writing process involves the creation of D_i -to- Dh_i “maps” that relate the geometrical features of the 0-halo sub-domains to their counterparts in the extended sub-domains Dh_i . It also implies generating new lists to store the mesh-dependent variables of the first layer of halo cells L_1 . These variables are indeed required by the WENO scheme to perform high-order calculations of the flux at the inter-processor boundaries.

At run time, the level set transport is performed on the 0-halo sub-domains. However, as the re-written mesh-dependent variables refer to cells in the extended

sub-domains, an intermediate “virtual field” $\tilde{\varphi}$ is used to perform the operations of the WENO scheme. This field is of the size of the extended sub-domains Dh_i and is populated with the level set scalar. The halo cells in $\tilde{\varphi}$ are updated, through MPI transfers, according to the level set field in the neighbouring 0-halo sub-domains. The level set field φ is then calculated in the 0-halo sub-domain from the virtual field $\tilde{\varphi}$.

As implemented in the code, the n-halo parallelisation involves the following steps:

1. Decomposition of the global domain D into extended sub-domains Dh_i .
2. Calculation of the mesh-dependent variables on the extended sub-domains. At this step, both these pre-processing variables and some geometrical characteristics of Dh_i are written out to construct the D_i -to- Dh_i maps at a later stage.
3. Decomposition of the global domain D into 0-halo sub-domains D_i .
4. Construction of the D_i -to- Dh_i maps that relate the 0-halo sub-domains to the extended sub-domains.
5. Re-writing of the pre-processing variables for D_i using the D_i -to- Dh_i maps.
6. Parallel run of the computation using a virtual field $\tilde{\varphi}$ to transport the level set field φ with high-order accuracy.

Droplet transfer in parallel

Just like the rest of the modelling capability, the transfer of droplets has been implemented in parallel. The parallel run of the algorithm on several processors involves two specific cases: droplets spanning on several sub-domains and the re-initialisation of the level set field φ on all cores simultaneously if a transfer of droplet has occurred somewhere in the domain.

As mentioned in Section 8.2.4, droplets spanning on several processors are ignored by the selection algorithm. Therefore, these drops remain in the level set formulation until they are convected well inside a sub-domain. The test that handles this case checks that none of the cells of the droplet considered are on a processor boundary. This test is made computationally efficient by producing, at pre-processing, lists of inter-processor boundary cells for each sub-domain.

Depending on the flow field simulated, droplets may be removed from the level set field in some sub-domains and not in others. However, it is crucial to synchronise the re-initialisation of the hyperbolic tangent profile on all the processors or the parallel computation will crash. This problem is mitigated by the creation of a list of booleans storing the state of each sub-domain. Whenever, a droplet is removed from φ in given sub-domain, its state is set to 1 — signifying the need for re-initialisation — and an all-to-all transfer of the sub-domain states is performed. The re-initialisation then occurs on each sub-domain as long as there is at least one state set to 1 in the list.

8.4 Performance of the method on basic two-phase flow problems

The calculations presented in this sub-section are typical tests that assess the performance of the multiphase modelling capability. In particular, we present here the results obtained by our method on two test cases: the Rayleigh-Taylor instability and the falling drop in a pool. Through these simulations we illustrate four features of our method:

- Accuracy,
- Robustness,
- Ability to run on different types of mesh,

- Performance in parallel.

8.4.1 Rayleigh-Taylor instability

The settings for the simulation of the Rayleigh-Taylor instability are given Section 4.2.2. For this test case, the transport of the volume fraction φ was performed with our novel modelling capability (*RCLSFoam*) using the linear high-order scheme and setting the interface thickness parameter to $\epsilon = 0.5\overline{\Delta x}$. This highlights the robustness of our method which copes with a density ratio of 7.4 (see Table 4.6) without resorting to a WENO treatment of the discontinuity.

Ability to capture the physics

The relative performance of *RCLSFoam* with respect to OpenFOAM's multiphase flow solver (*interFoam*) is presented in Figure 8.10. This figure shows the solution obtained by the two codes on the same triangular mesh made of 9234 cells, for six different times in $[0; 1.5]$ seconds.

The comparison of the interface predicted by the two solvers for the first three times ($t = 0.7s, 0.8s, 0.9s$) demonstrates the superior performance of *RCLSFoam* on asymmetrical meshes. Indeed, while the solution predicted by our modelling capability remains close to the reference solution (see Gerris' prediction on 256×1024 mesh Figure 4.5), OpenFOAM's predictions are unrealistically asymmetrical. This is probably due to the lower order of the numerical schemes used by *interFoam* (smaller stencils) to transport the volume fractions. It suggests that *interFoam* is not appropriate on general unstructured grids.

The solution predicted by *interFoam* for the last three times ($t = 1.25s, 1.4s, 1.45s$) illustrates a fundamental flaw of this solver. As already mentioned in Section 4.2.2, *interFoam* produces non-physical wiggles in the stem of the mushroom-shaped structure formed by the penetration of the denser phase into the lighter phase. Figure 8.10 shows the build-up of these non-physical interfacial oscillations

until they break the mushroom-shaped structure (see results for $t = 1.45s$).

Through various tests, we have identified that these non-physical features were due to the compression flux implemented by Weller to maintain the sharpness of the interface. However, this compression flux cannot be simply removed from *interFoam* or Weller's modelling tool would fall apart. As our modelling capability maintains the interface sharp through the solution of the re-initialisation equation (see Chapter 7), the Rayleigh-Taylor instability is correctly simulated. The ability of *RCLSFoam* to model the Rayleigh-Taylor instability is particularly relevant to the simulation of the atomisation process as this instability plays a significant role in the primary breakup (see Chapter 2).

Handling of hybrid meshes

RCLSFoam's ability to perform accurately on different types of grid is demonstrated by comparing the solutions obtained on a triangular and a hybrid (triangles-rectangles) mesh of similar size. Figure 8.11 shows the two meshes considered for this study: a triangular mesh of 9234 cells and a hybrid grid of 8580 elements. The rectangles of the hybrid mesh are distributed all along the width, in the middle portion of the domain — where the interface is initialised — and the rest of the domain is made of triangular elements.

Figure 8.11 presents the volume fractions and interface predicted by *RCLSFoam* on the two meshes for five different times in $[0; 1]$ second. As expected the results obtained are very similar and, thanks to the high-order scheme implemented, the method copes very well with the change of mesh type in the hybrid grid.

Small differences for the solution at time $t = 1s$ can be perceived between the two computations: the interface obtained on the hybrid mesh is slightly more symmetrical. This is due to the initialisation of the volume fraction field: whereas the sine interface is set on a perfectly symmetrical Cartesian mesh for the hybrid grid, it is initialised on a non-symmetrical mesh for the triangular grid. As the initialisation of

the interface is critical to the symmetry of the solution, the results obtained on the hybrid mesh are more symmetrical. It is also worth noting that the progression of the mushroom-shaped structure in the triangular part of the hybrid mesh does not affect its symmetry. This also illustrates how one can take advantage of the ability to run on hybrid meshes and further highlights the relevance of this capability.

Mass conservation

In Chapter 6 and 7, a theoretical justification was given to attest that our numerical transport of the smeared out liquid volume fraction leads to an exact mass conservation. All the numerical experiments that we have performed in serial have confirmed this assertion. In particular, we reported in Figure 8.12 the total mass error observed for the simulation of the Rayleigh-Taylor instability on a triangular mesh. These results confirmed that the RCLS conserves mass to machine accuracy in serial.

Performance in parallel

As already mentioned in Section 8.3, OpenFOAM adopts a 0-halo approach to parallelisation. However, our modelling capability transports the volume fractions with a high-order numerical scheme, such that a 0-halo parallel run with *RCLSFoam* leads to non-physical solutions in the vicinity of the inter-processor boundaries. To mitigate this issue we have implemented a n-halo parallelisation in OpenFOAM.

Figure 8.12 compares the predictions of the 0-halo and n-halo parallel computations of the Rayleigh-Taylor instability with the corresponding serial run. All three calculations have been performed on the same triangular mesh made of 9234 elements and the predicted interface is given for five different times in $[0; 1]$ second. In order to highlight any disturbance caused by the parallelisation, the domain is decomposed in four sub-domains by cutting half-way along the horizontal and vertical directions. The domain decomposition associated to each parallel approach is

overlying the solution at time $t = 0.2s$ in Figure 8.12.

The 0-halo parallel run presents non-physical features in the vicinity of the inter-processor boundaries: distorted interface and inclusion of light phase in the cap of the mushroom-shaped structure (see interface at times $t = 0.8s, 0.9s, 1s$). As expected, the n-halo parallel run is free from non-physical features and matches almost exactly the interface predicted by the serial run.

Through various tests, we have identified that the discrepancy between the serial and the n-halo parallel run comes from a fundamental flaw in the parallel handling of the pressure-velocity coupling in OpenFOAM. Unfortunately, this flaw affects the mass conservation properties of the method in parallel.

NB: The n-halo parallel run of the RCLS method alone matches its serial counterpart to machine accuracy.

As both *interFoam* and *RCLSFoam* conserve mass to machine accuracy when running in serial, the impact of OpenFOAM's flaw can be measured in terms of mass variation over a computation. In particular, we have measured the absolute variation of mass over a run of the Rayleigh-Taylor instability between the times $t = 0s$ and $t = 1.5s$: $|\Delta m_0^{1.5}|$ (see the right most column in Figure 8.12).

The calculation of $|\Delta m_0^{1.5}|$ confirms that *RCLSFoam* conserves mass to machine accuracy in serial. However, with a 0-halo parallelisation, the absolute mass variation over the whole computation is $O(10^{-3})$. Thanks to the implementation of a n-halo parallel approach, this mass variation is reduced to $O(10^{-8})$. As a frame of reference, the parallel simulation of the Rayleigh-Taylor instability with *interFoam* leads to an absolute variation of mass of: $|\Delta m_0^{1.5}| = 1.94 \times 10^{-6}$. With a mass error two orders of magnitude smaller than *interFoam*'s, once again, our modelling capability outperforms Weller's code.

8.4.2 Falling drop in a pool

The settings for the simulation of the falling drop in a pool are given Section 4.2.2. For this test case, the transport of the volume fraction φ was performed with *RCLS-Foam* using the WENO scheme and setting the interface thickness parameter to $\epsilon = 0.5\overline{\Delta x}$. This computation is notoriously challenging as it involves a density ratio of 816 (20 times larger than the density ratio encountered in aero-engines, see tables 4.7 and 4.9).

For this computation, the solution of the pressure-velocity coupling is performed using a Crank-Nicholson (CN) temporal discretisation and the resulting systems of algebraic equations are solved with the generalised geometric-algebraic multi-grid solver (GAMG). Although this choice of numerics generally improves the accuracy (CN) and the speed (GAMG) of the computation, it significantly reduces the stability margin of the numerical method. As a result, the successful simulation of the falling drop with such numerics is an achievement in itself. Indeed, no stable computation were obtained for this test case with *interFoam* using either of these numerical methods. This further highlights the superior robustness of our modelling capability.

Figure 8.13 presents the volume fractions and the interface predicted by *RCLS-Foam* on a 128×256 Cartesian mesh for five different times in $[0; 0.55]$ millisecond. It can be seen in the Figure 8.13, that the *RCLS-Foam* copes remarkably well with the very large density ratio and that the results match closely the solution provided by Gerris on the same mesh (see Figure 4.8).

To conclude, in this section we have demonstrated that our modelling capability captures the physics well on any kind of unstructured grid. We also showed that *RCLS-Foam* conserves mass to machine accuracy in serial and to very high accuracy in parallel. In addition, we have established that this multiphase modelling

tool is very robust. Finally, it follows from these numerical tests that *RCLSFoam* outperforms *interFoam* in terms of accuracy, mass conservation and robustness.

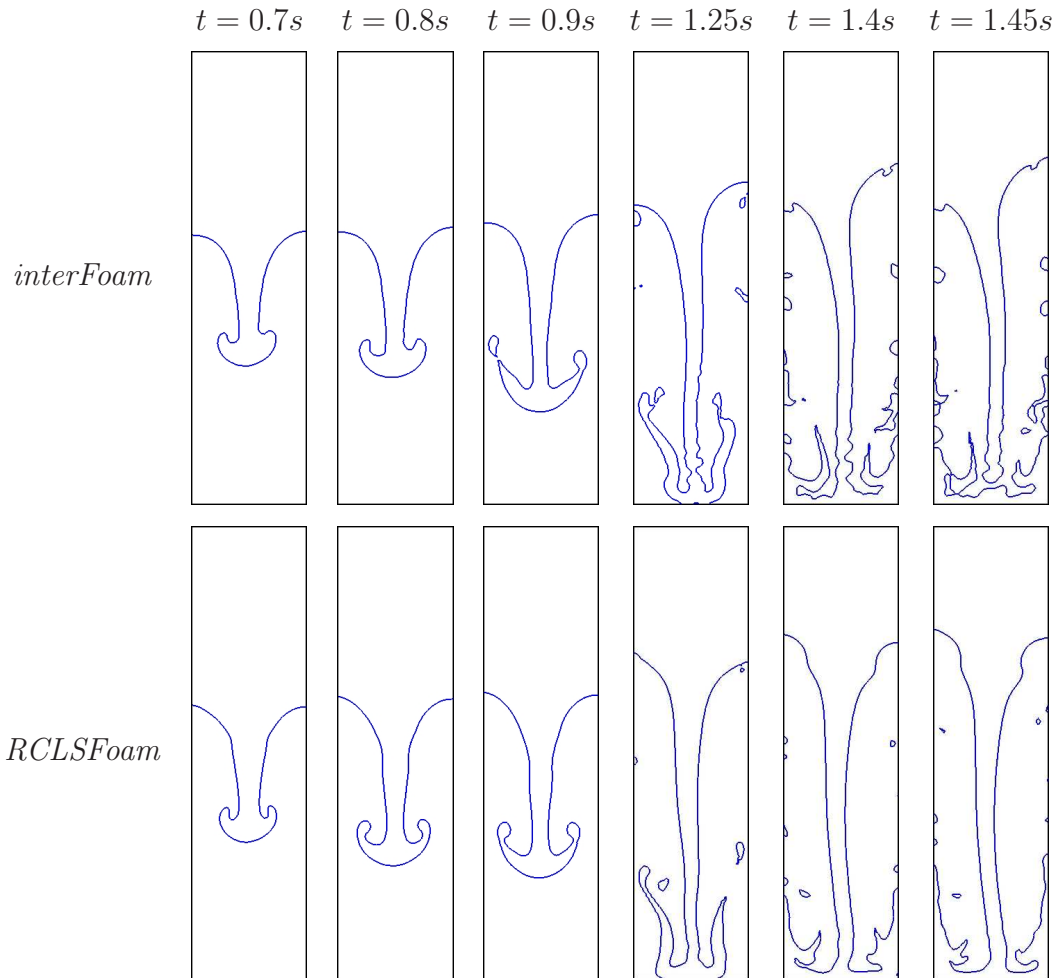


Figure 8.10: Interface predicted by *RCLSFoam* for the Rayleigh-Taylor instability
— Comparison *RCLSFoam* vs. *interFoam*

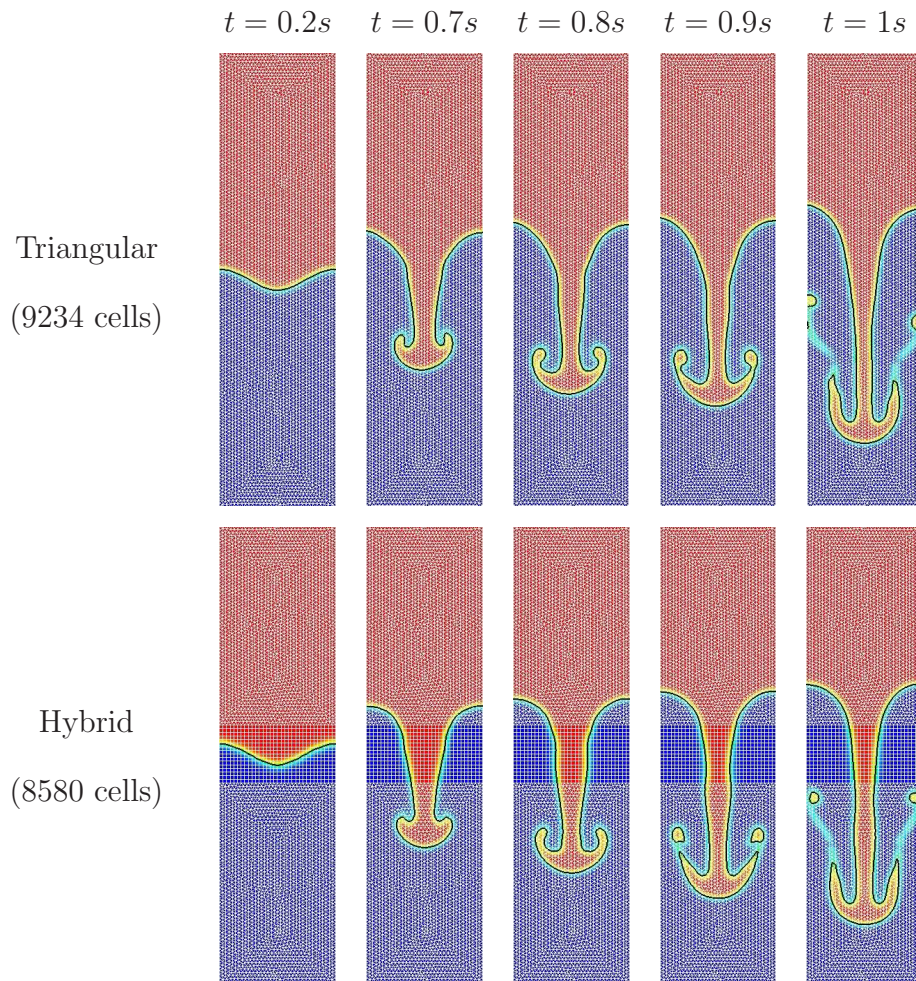


Figure 8.11: Volume fractions and interface (in black) predicted by *RCLSFoam* for the Rayleigh-Taylor instability — Comparison hybrid vs. triangular mesh (meshes overlaying the contours in white)

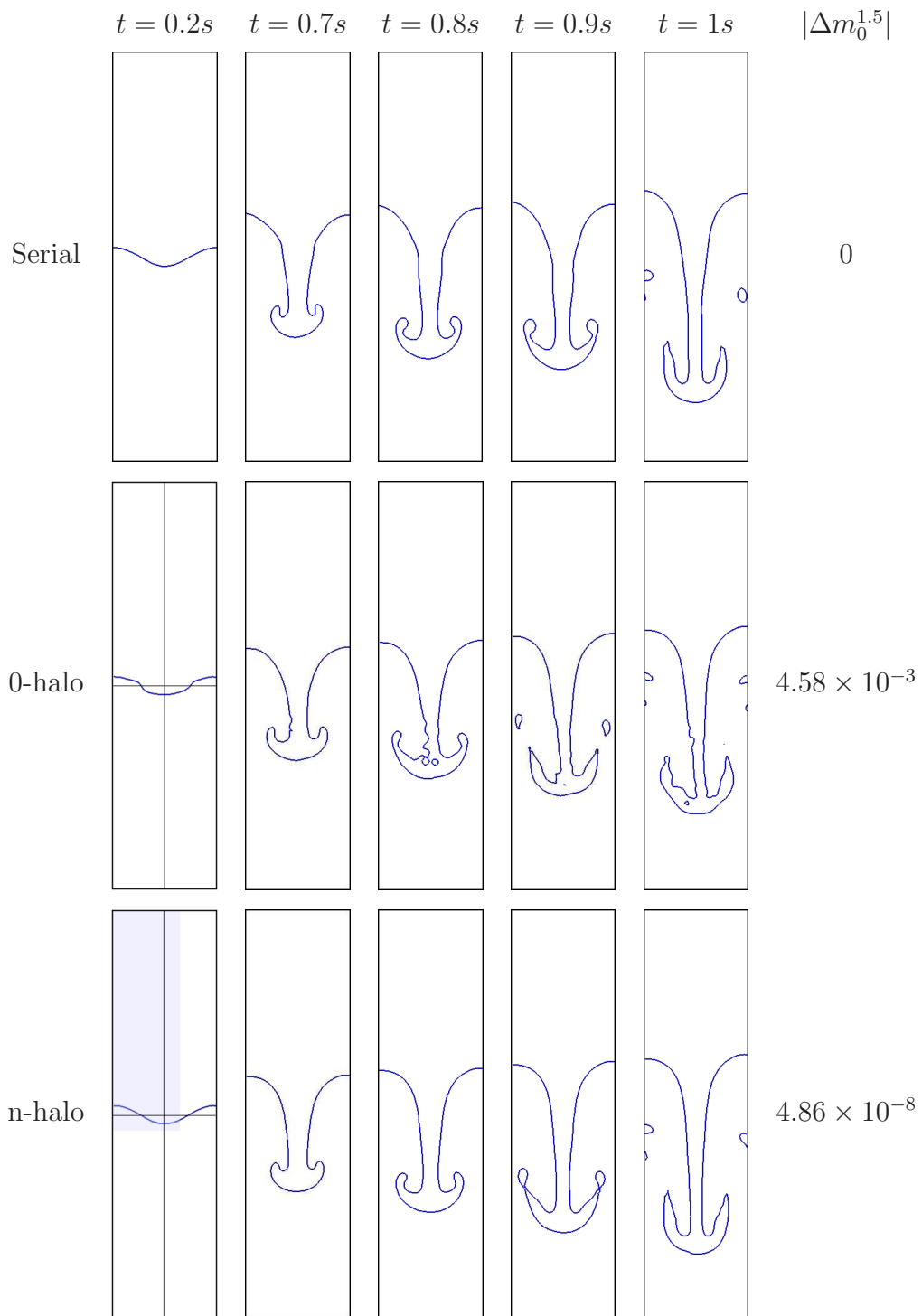


Figure 8.12: Interface predicted by *RCLSfoam* for the Rayleigh-Taylor instability — Comparative performance of the 0-halo and n-halo parallel approaches

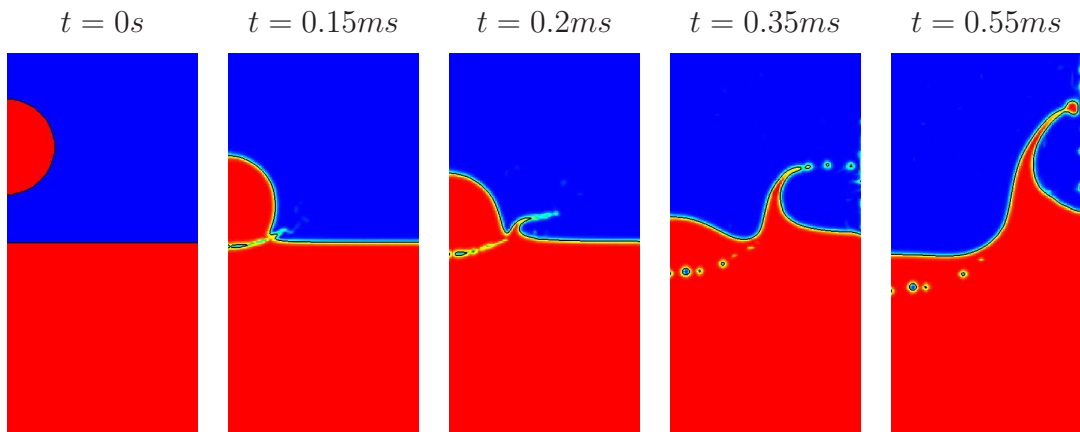


Figure 8.13: Volume fractions and interface (in black) predicted by *RCLSFoam* for the falling drop in a pool — Time $t = 0s, 0.15ms, 0.2ms, 0.35ms, 0.55ms$

Chapter 9

Simulation of liquid sheet breakup

The modelling capability as developed is applicable to both DNS and LES of atomisation. As our methodology relies on an implicit interface description method, the breakup phenomenon is handled implicitly (no pinch off models) and its length scale is of the order of the local grid size. As argued by Gorokhovski and Herrmann in [71], although this makes the topology changes grid dependent, it does not significantly affect the droplets larger than a few cell volumes.

To perform DNS, sufficient grid resolution has to be provided to capture the Kolmogorov length scales in both phases, η_l and η_g , and the smallest liquid structure produced ζ (typically drops at $We = 10$). As this is still infeasible in practice for typical aero-engine conditions, it is necessary to resort to a “quasi-DNS/LES” methodology [71].

In this chapter we present the sheet breakup of liquid fuel in co-flowing gas, simulated using a quasi-DNS/LES methodology. The quasi-DNS/LES formulation adopted for this calculation is described in Section 9.1, the settings of the computation are given in Section 9.2 and the results obtained with the modelling capability (*lesRCLSFoam*) are presented in Section 9.3.

9.1 Quasi-DNS/LES formulation

This numerical framework involves applying a single-phase LES formulation in both phases and extending it to the region of the interface. Due to the lack of established models, this approach neglects the sub-grid scale effects associated with the presence of the interface.

9.1.1 Filtering

As mentioned in Chapter 3, LES only resolves the larger scales of the turbulence while the effect of the smaller scales are modelled. Therefore, the flow variables (\mathbf{u}, p) are decomposed into filtered components $(\bar{\mathbf{u}}, \bar{p})$ and residual components (\mathbf{u}', p') , according to:

$$\mathbf{u} = \bar{\mathbf{u}} + \mathbf{u}' \quad (9.1)$$

$$p = \bar{p} + p' \quad (9.2)$$

Filter definition

The separation of the resolved scales from the sub-grid scales is performed by filtering the flow variables [123]. Introducing the filter function G_Δ , its associated filter width Δ (proportional to the smallest length scale resolved) and the computational domain Ω , this filtering operation is defined by the following convolution [180, 211]:

$$\bar{\mathbf{u}}(\mathbf{x}, t) = G_\Delta * \mathbf{u} = \iiint_{\Omega} G_\Delta(\mathbf{r}, \mathbf{x}) \mathbf{u}(\mathbf{x} - \mathbf{r}, t) \, d\mathbf{r} \quad (9.3)$$

where the filter kernel satisfies the normalisation condition:

$$\iiint_{\Omega} G_\Delta(\mathbf{r}, \mathbf{x}) \, d\mathbf{r} = 1 \quad (9.4)$$

The filter function G_Δ defined above satisfies the following properties [211]:

1. Conservation of constants (equivalent to (9.4)). For a constant vector \mathbf{g} this property reads:

$$\overline{\mathbf{g}} = \mathbf{g} \quad (9.5)$$

2. Linearity. Introducing two vectorial space-time variables \mathbf{u} and \mathbf{v} , this condition reads:

$$\overline{\mathbf{u} + \mathbf{v}} = \overline{\mathbf{u}} + \overline{\mathbf{v}} \quad (9.6)$$

3. Commutation with differentiation. As G_Δ is independent of time t (see (9.3)), the filtering operation and the derivative with respect to time commute:

$$\overline{\frac{\partial \mathbf{u}}{\partial t}} = \frac{\partial \overline{\mathbf{u}}}{\partial t} \quad (9.7)$$

In the case of an homogeneous filter: $G_\Delta(\mathbf{r}, \mathbf{x}) = G_\Delta(\mathbf{r})$, the filtering operation and the derivative with respect to the position commute:

$$\overline{\frac{\partial u_i}{\partial x_j}} = \frac{\partial \overline{u_i}}{\partial x_j} \quad (9.8)$$

Although the decomposition of the flow variables into resolved and residual components is analogous to a Reynolds decomposition, it is worth noting that the filtered components $(\overline{\mathbf{u}}, \overline{p})$ are random fields and that the filtered residual components $(\overline{\mathbf{u}'}, \overline{p'})$ are not systematically null. Therefore, in general, a filter (even homogeneous) is not a ‘‘Reynolds operator’’ [211] and:

$$\overline{\overline{\mathbf{u}}} \neq \overline{\mathbf{u}} \quad (9.9)$$

$$\overline{\overline{\mathbf{u}'}} \neq \mathbf{0} \quad (9.10)$$

$$\overline{\overline{\mathbf{u}\mathbf{u}}} \neq \overline{\mathbf{u}}\overline{\mathbf{u}} \quad (9.11)$$

Non-uniform filter width

A filter can only be considered homogeneous on a uniform grid. In the case of unstructured grids, the filter width is a function of space $\Delta = \Delta(\mathbf{x})$ and a commutation error appears. By choosing carefully the filter kernel G_Δ , Vasilyev et al. showed that

it is possible to control the order of this error on unstructured grids [82, 147, 265]. However, this approach involves explicit filtering which can be cumbersome and computationally expensive [43].

After Villiers, we choose to minimise the commutation error by smoothing the distribution of filter widths [43]. For each element E_i of the mesh, a “cell-derived” filter width Δ_{E_i} can be calculated as a function of the cell volume $|E_i|$:

$$\Delta_{E_i} = \sqrt[3]{|E_i|} \quad (9.12)$$

Let us note Δ_i the filter width of E_i . Instead of setting directly $\Delta_i = \Delta_{E_i}$, the principle of the smoothing method is to increase the filter width of the cells neighbouring larger control volumes [43]. For this purpose, a fixed gradient for the smoothed distribution is defined — based on a coefficient C_Δ — such that $\Delta_i \geq \Delta_{E_i}$. Introducing the set of cells neighbouring E_i : $\mathcal{N}_i = \bigcup_j E_j$, the smoothing scheme reads:

$$\Delta_i = \max_{j \in \mathcal{N}_i} \left(\Delta_{E_i}, \frac{\Delta_{E_j}}{C_\Delta} \right) \quad (9.13)$$

As recommended by Villiers, we have set the smoothing coefficient to:

$$C_\Delta = 1.1 \quad (9.14)$$

In his numerical experiments, Villiers observed that the smoothing approach was sufficient to obtain relatively good results at a fairly low cost.

9.1.2 Filtered Navier-Stokes equations

The convolution of (8.1) and (8.2) with an homogeneous kernel filter G_Δ provides the following filtered incompressible Navier-Stokes equations:

$$\nabla \cdot \bar{\mathbf{u}} = 0 \quad (9.15)$$

$$\frac{\partial (\rho \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = -\nabla \bar{p} + \rho \mathbf{g} + \sigma \kappa \nabla \varphi + \nabla \cdot (2\mu \bar{\mathbf{D}} - \rho \mathcal{T}) \quad (9.16)$$

where:

- The filtered rate-of-strain tensor $\overline{\mathcal{D}}$ reads:

$$\overline{\mathcal{D}} = \frac{1}{2} \left(\nabla \overline{\mathbf{u}} + (\nabla \overline{\mathbf{u}})^T \right) \quad (9.17)$$

- \mathcal{T} is the residual-stress tensor:

$$\mathcal{T} = \overline{\mathbf{u} \otimes \mathbf{u}} - \overline{\mathbf{u}} \otimes \overline{\mathbf{u}} \quad (9.18)$$

- Following the principle of the quasi-DNS/LES approach, the sub-grid scale contributions of the capillary forces $(\mathbf{f}_{\text{cap}})_{SGS}$ have been neglected:

$$(\mathbf{f}_{\text{cap}})_{SGS} = \mathbf{0} = \sigma \kappa \nabla \varphi - \overline{\sigma \kappa \nabla \varphi} \quad (9.19)$$

$$\overline{\sigma \kappa \nabla \varphi} = \sigma \kappa \nabla \varphi \quad (9.20)$$

Leonard proposed the following decomposition of the residual-stress tensor:

$$\begin{aligned} \mathcal{T} &= (\overline{\mathbf{u} \otimes \mathbf{u}} - \overline{\mathbf{u}} \otimes \overline{\mathbf{u}}) + (\overline{\mathbf{u} \otimes \mathbf{u}'} + \overline{\mathbf{u}' \otimes \mathbf{u}}) + (\overline{\mathbf{u}' \otimes \mathbf{u}'}) \\ &= \mathcal{L} + \mathcal{C} + \mathcal{R} \end{aligned} \quad (9.21)$$

where the Leonard-stress tensor \mathcal{L} represents the “interactions among the large scales”, the cross-stress tensor \mathcal{C} reflects the “interactions between large and small scales” and the SGS Reynolds-stress tensor \mathcal{R} stands for “the interactions between sub-grid scales” [211].

9.1.3 Residual kinetic energy

Filtering the kinetic energy field $E = \frac{1}{2} \mathbf{u} \cdot \mathbf{u}$ produces \overline{E} :

$$\overline{E} = \frac{1}{2} \overline{\mathbf{u} \cdot \mathbf{u}} \quad (9.22)$$

Decomposing \overline{E} into the kinetic energy of the filtered velocity field E_f and the residual kinetic energy k_r leads to [180]:

$$\overline{E} = E_f + k_r \quad (9.23)$$

where k_r is expressed as:

$$k_r = \frac{1}{2}\overline{\mathbf{u} \cdot \mathbf{u}} - \frac{1}{2}\overline{\mathbf{u}} \cdot \overline{\mathbf{u}} \quad (9.24)$$

Noting that $k_r = \frac{1}{2}\text{tr}(\mathcal{T})$, the decomposition of the residual-stress tensor into an isotropic part and a deviatoric part \mathcal{T}_D can be expressed as follows:

$$\mathcal{T} = \frac{2}{3}k_r\mathcal{I} + \mathcal{T}_D \quad (9.25)$$

where \mathcal{I} is the unit tensor.

9.1.4 Sub-grid scale modelling

For this computation, we have chosen to model the sub-grid scale stress tensor \mathcal{T} with the constant coefficient one-equation eddy-viscosity model (OEEVM) proposed by Yoshizawa [281].

Eddy-viscosity assumption

The OEEVM is based on Boussinesq's eddy-viscosity concept which postulates that the mechanism governing the transfer of energy from the resolved scales to the residual scales is analogous to the mechanism driving the molecular diffusion. With this hypothesis, the deviatoric part of the residual-stress tensor \mathcal{T}_D is assumed locally aligned with the deviatoric part of the filtered rate-of-strain tensor \mathcal{D}_D and the normal stresses (considered isotropic) are calculated from the residual kinetic energy k_r [64]. Introducing the eddy-viscosity of the residual motions ν_r , Boussinesq's assumption leads to the following expression for the residual-stress tensor:

$$\mathcal{T} = \frac{2}{3}k_r\mathcal{I} - 2\nu_r\overline{\mathcal{D}_D} \quad (9.26)$$

However, as the flow is considered incompressible, the filtered dilatation is null:

$$\overline{d} = \text{tr}(\overline{\mathcal{D}}) = \nabla \cdot \overline{\mathbf{u}} = 0 \quad (9.27)$$

Such that:

$$\overline{\mathcal{D}} = \overline{\mathcal{D}_D} + \frac{1}{3}\overline{d}\mathcal{I} = \overline{\mathcal{D}_D} \quad (9.28)$$

It follows that SGS eddy-viscosity type models for incompressible flows express the residual-stress tensor as the following function of k_r and ν_r :

$$\mathcal{T}(k_r, \nu_r) = \frac{2}{3}k_r\mathcal{I} - 2\nu_r\overline{\mathcal{D}} \quad (9.29)$$

The algebraic eddy-viscosity model of Smagorinsky [234] calculates directly the parameters k_r and ν_r from the characteristic filtered rate of strain: $\overline{\mathcal{D}} = (2\overline{\mathcal{D}} : \overline{\mathcal{D}})$ by assuming that the rate of production of the residual kinetic energy is exactly balanced by the rate of dissipation of kinetic energy. As this balance breaks down near walls and in jets and wakes, Yoshizawa proposed a model equation for the residual kinetic energy k_r .

Transport equation for the residual kinetic energy

The exact transport equation for k_r (TK) can be obtained by combining the momentum equation (ME) (given by (8.2)) and its filtered counterpart (\overline{ME}) (given by (9.16)) in the following way [41]:

$$(TK): \quad \overline{\mathbf{u} \cdot (ME)} - \overline{\mathbf{u}} \cdot (\overline{ME})$$

Neglecting the terms related to the capillary forces in (TK) leads to the following equation for the residual kinetic energy [41]:

$$\underbrace{\frac{\partial k_r}{\partial t} + \nabla \cdot (\overline{\mathbf{u}}k_r)}_{Convection} = \underbrace{-\nabla \cdot \left(\frac{1}{2} \overline{(\mathbf{u} \cdot \mathbf{u})\mathbf{u}} - \overline{E}\overline{\mathbf{u}} + \frac{1}{\rho} \overline{p\mathbf{u}} - \frac{1}{\rho} \overline{p}\overline{\mathbf{u}} - \overline{\mathbf{u}} \cdot \mathcal{T} \right)}_{Diffusion} + \nabla \cdot (\nu \nabla k_r) - \underbrace{\nu (\overline{\nabla \mathbf{u}} : \overline{\nabla \mathbf{u}} - \nabla \overline{\mathbf{u}} : \nabla \overline{\mathbf{u}})}_{Dissipation} - \underbrace{\overline{\mathcal{D}} : \mathcal{T}}_{Production} \quad (9.30)$$

Yoshizawa derives a model transport equation for the residual kinetic energy by using the statistical results from the direct-interaction approximation [114–116]. As this statistical approach assumes the distinct separation of the grid-scales mean motions and the sub-grid scale fluctuating motions [280, 282], the Leonard-stress tensor \mathcal{L} and the cross-stress tensor \mathcal{C} are neglected.

Noting D_{k_r} the non-molecular diffusion term in (9.30) and ε_{k_r} the dissipation term in (9.30), Yoshizawa's statistical framework leads to the following models for D_{k_r} and ε_{k_r} :

$$D_{k_r} = \nabla \cdot (\nu_r \nabla k_r) \quad (9.31)$$

$$\varepsilon_{k_r} = \frac{C_\varepsilon k_r^{\frac{3}{2}}}{\Delta} \quad (9.32)$$

with the following expression for the eddy-viscosity of the residual motions:

$$\nu_r = C_\nu k_r^{\frac{1}{2}} \Delta \quad (9.33)$$

As a result, the model transport equation for the residual kinetic energy reads:

$$\frac{\partial k_r}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} k_r) = \nabla \cdot ((\nu_r + \nu) \nabla k_r) - \varepsilon_{k_r} - \bar{\mathcal{D}} : \mathcal{T} \quad (9.34)$$

The same model equation for the transport of k_r was derived by Schumann using dimensional analysis [220].

Yoshizawa recommends the model parameters $C_\nu \simeq 0.05$ and $C_\varepsilon \simeq 1$. Building upon the successful application of this SGS model to the simulation of diesel jet atomisation by Villiers et al., we will use the following values for C_ν and C_ε [44]:

$$C_\nu = 0.07 \quad (9.35)$$

$$C_\varepsilon = 1.05 \quad (9.36)$$

Performance of the OEEVM

As intended, this model performs better than the Smagorinsky model in regions of the flow where the balance between production and dissipation breaks down: wakes, jets and near-wall flows. And wherever the production-dissipation equilibrium is valid the OEEVM revert to the Smagorinsky model [64].

However, as the eddy-viscosity concept assumes the isotropy of the normal residual stresses, the OEEVM requires a high mesh density in near-wall regions to capture

the flow accurately [63]. Besides, as the eddy viscosity of the residual motions ν_r is by definition always positive in Yoshizawa's model (9.33), the OEEVM does not account for the transfer of energy from the sub-grid scales to the resolved scales (backscatter).

To summarise, Yoshizawa's model performs relatively well on the types of flow encountered in atomisation problems while remaining significantly cheaper than more advanced SGS models [64].

9.1.5 Quasi-DNS/LES equations

The set of equations solved in this numerical framework is recapitulated below:

Transport of the liquid volume fraction

$$\frac{\partial \varphi}{\partial t} + \nabla \cdot (\varphi \bar{\mathbf{u}}) = 0$$

Filtered incompressible Navier-Stokes equations

$$\nabla \cdot \bar{\mathbf{u}} = 0$$

$$\frac{\partial (\rho \bar{\mathbf{u}})}{\partial t} + \nabla \cdot (\rho \bar{\mathbf{u}} \otimes \bar{\mathbf{u}}) = -\nabla \bar{p} + \rho \mathbf{g} + \sigma \kappa \nabla \varphi + \nabla \cdot (2\mu \bar{\mathcal{D}} - \rho \mathcal{T})$$

with: $\bar{\mathcal{D}} = \frac{1}{2} \left(\nabla \bar{\mathbf{u}} + (\nabla \bar{\mathbf{u}})^T \right)$, $\mathcal{T} = \frac{2}{3} k_r \mathcal{I} - 2\nu_r \bar{\mathcal{D}}$, $\nu_r = C_\nu k_r^{\frac{1}{2}} \Delta$

Transport of the residual kinetic energy

$$\frac{\partial k_r}{\partial t} + \nabla \cdot (\bar{\mathbf{u}} k_r) = \nabla \cdot ((\nu_r + \nu) \nabla k_r) - \varepsilon_{k_r} - \bar{\mathcal{D}} : \mathcal{T}$$

with: $\varepsilon_{k_r} = \frac{C_\varepsilon k_r^{\frac{3}{2}}}{\Delta}$

9.2 Settings of the computation

Previous numerical studies of atomisation have highlighted the sensitivity of the computation to boundary conditions (see Chapter 5). This section details the choices made in this demonstration exercise. In particular, the mechanical and physical boundary conditions are given in Section 9.2.1 and in Section 9.2.2 we present a parametric study of the RCLS method which leads to the choice of settings for the computation.

9.2.1 Domain and material properties

In Chapter 4, we reproduced the results obtained by Villiers et al. for the simulation of Diesel jet atomisation [44]. We then extended this approach (similar physical properties and domain dimensions) to the computation of liquid sheet breakup in co-flow (see Section 4.3.2).

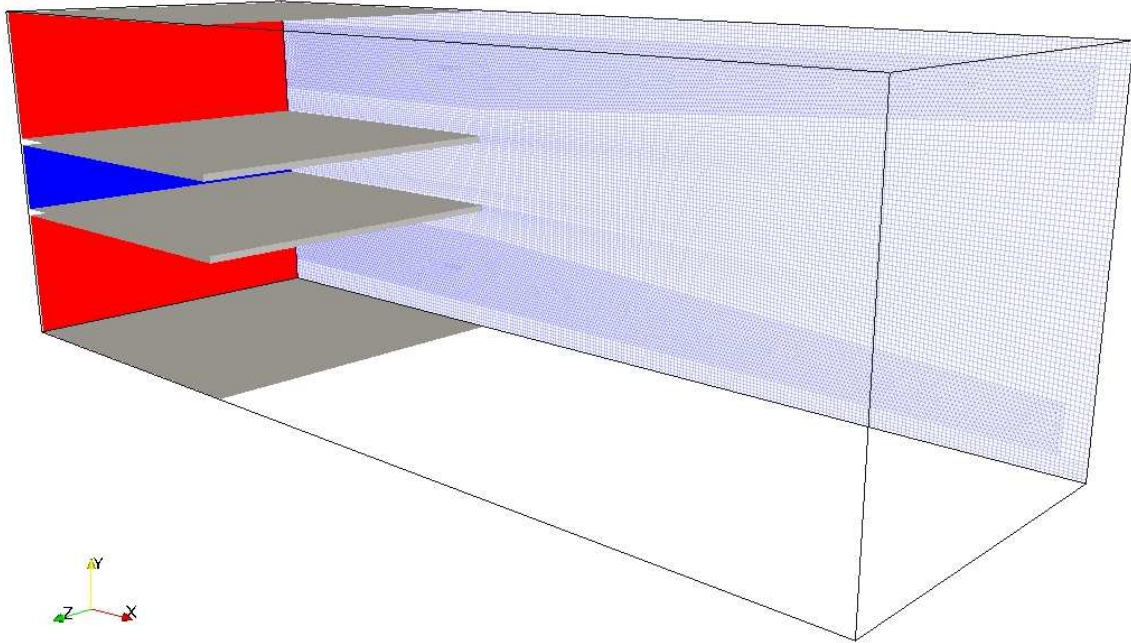
In order to demonstrate the modelling capability on a test case reproducing the breakup mechanisms observed at aero-engine conditions, we have chosen to compute the flat sheet breakup at $We = 9300$.

Computational domain

The computational domain generated for this simulation is a cuboid of $3 \times 1 \times 1$ mm^3 , meshed with 2.36×10^6 cells. In order to demonstrate the capability on general polyhedral meshes, the grid is made of 8.07×10^5 hexahedra, 1.5×10^6 tetrahedra and 5.4×10^4 pyramids (see bottom picture in Figure 9.1).

The gas and fuel injection channels are modelled — they span over $1mm$ in the longitudinal direction — and the thickness of the plates separating the injection channels is resolved (see top picture in Figure 9.1). Both of these features have proven to be of significant importance in the numerical simulation of primary breakup (see [65]). The geometrical details of the injection configuration can be found in the “Flat sheet” column of Table 4.11.

Domain and boundary conditions



Mesh along the centreplane ($z = 0$)

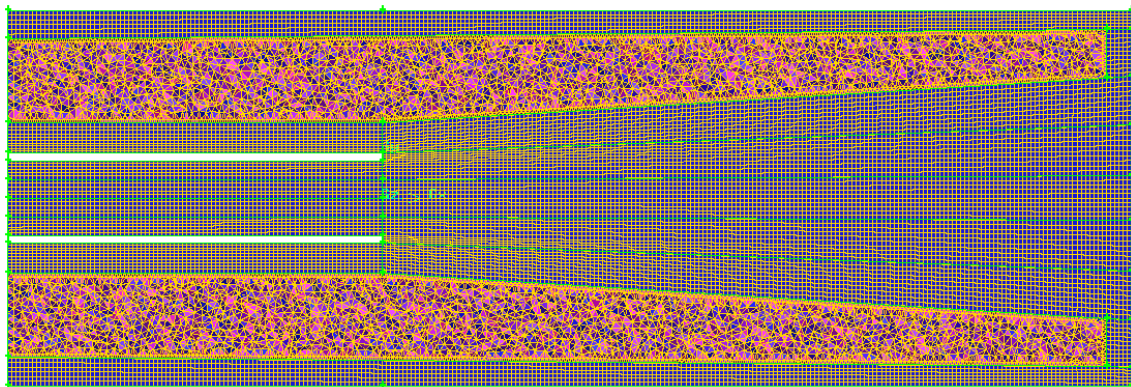


Figure 9.1: Computational domain for the simulation of atomisation — In the top half: the mesh of the back plane (edges in blue) and the boundaries (walls in grey, fuel inlet in blue and gas inlets in red) are pictured

The front and back planes — respectively defined by $z = 0.5$ and $z = -0.5$ (see top picture in Figure 9.1) — are set as symmetric boundary condition. The plane at

$x = 0$ is the inlet of the computational domain. In the central part of this plane — between the two plates — the fuel ($\varphi = 1$) is injected at a speed of $2ms^{-1}$. Above and below the fuel injection channel, the gas ($\varphi = 0$) enters the domain at a speed of $40ms^{-1}$. The rest of the boundaries are defined as inlet/outlet boundary conditions.

In order to take advantage of the ability to run on hybrid meshes, structured hexahedral grids have been fitted around walls. Similarly, to improve the accuracy of the atomisation modelling, the regions where the breakup is expected to occur have been meshed with hexahedral cells. Because of the instability of the inlet/outlet boundary conditions in OpenFOAM, these boundaries have also been meshed with hexahedra. The rest of the domain is composed of tetrahedra and pyramids. The pyramidal mesh consists of a single layer of cells performing the transition between the hexahedral and the tetrahedral meshes.

In the resulting mesh, the smallest edge length is $\Delta x_{min} = 5\mu m$ and the average characteristic length scale of the grid (as defined in (9.38)) is $\overline{\Delta x} = 8.4\mu m$.

Material properties

Although the material properties chosen for this study (see Table 4.9) differ slightly from those of an aero-engine, the physical quantities remain similar. The non-dimensional numbers related to this computation are given in Table 9.1 and the smallest length scales associated with the flow field are reported in Table 9.2.

Considering that a given length scale η is resolved if $\eta \geq 2\Delta x$, it can be seen in Table 9.2 that only the Kolmogorov length scale in the liquid phase η_{liq} is properly resolved by the mesh. This justifies the use of a sub-grid scale model for the turbulence in the gas phase.

Table 9.2 also suggests that the mesh is far from resolving the smallest liquid structures produced. This further highlights the need to develop sub-grid scale models for multiphase flows.

Table 9.1: Non-dimensional numbers associated to the flow simulated

| | |
|------------|-----------------------|
| Re_{liq} | 67 |
| Re_{gas} | 18800 |
| We | 9300 |
| Oh | 7.55×10^{-2} |

Table 9.2: Smallest length scale of the flow field — Assuming a turbulence intensity of 5% in the liquid and 10% in the gas

| | |
|---|-------------|
| Liquid phase Kolmogorov scale, η_{liq} | $81\mu m$ |
| Gas phase Kolmogorov scale, η_{gas} | $1.4\mu m$ |
| Droplet diameter at $We = 10$, $d_{We=10}$ | $0.22\mu m$ |
| Minimum mesh size, Δx_{min} | $5\mu m$ |
| Average mesh size, $\overline{\Delta x}$ | $8.4\mu m$ |

9.2.2 Choice of RCLS settings

The RCLS method involves the following parameters:

- The order of the polynomial reconstruction of the high-order scheme: r .
- The type of high-order scheme: linear or WENO.
- The coefficient controlling the thickness of the interface ϵ .
- The periodicity of the re-initialisation.

Although our numerical scheme can formally reach arbitrarily high order in space, in this demonstration of the modelling capability we limit ourselves to a third-order polynomial reconstruction: $r = 3$.

This leads to a third-order accurate linear scheme for the transport of the liquid volume fraction and in the case of a WENO reconstruction, the scheme is typically fourth-order accurate. In the interest of robustness and because atomisation problems involve large density ratios, we choose to use the WENO scheme for this calculation.

As mentioned in Section 7.2.4, the coefficient ϵ affects the stability of the method and — through the interface thickness — controls the resolution of the interface geometry. This coefficient is defined as a function of the mesh size (see (7.40)), such that on general unstructured meshes, ϵ varies from cell to cell. However, for our simulation, we will take ϵ uniform on the whole computational domain Ω :

$$\epsilon = C\overline{\Delta x} \tag{9.37}$$

where C is a constant coefficient.

Introducing the number of internal faces of the mesh: $N_{FI\Omega}$ and, for a given face F_l noting the cell centres of the owner and neighbour cells respectively O_l and N_l , $\overline{\Delta x}$ is given by:

$$\overline{\Delta x} = \frac{1}{N_{FI\Omega}} \sum_{l=1}^{N_{FI\Omega}} |\overrightarrow{O_l N_l}| \tag{9.38}$$

As demonstrated in Section 7.2.4, the method's stability requires $C \geq 0.5$ when ϵ is defined as a function of the local mesh size Δx .

Another parameter of conservative level set methods was introduced by Desjardins et al. in [47]: the periodicity of the re-initialisation. In numerical tests, the authors noted that an optimum value could be found for this parameter. In particular, they observed that re-initialising the level set profile less frequently (up to a certain point) would lead to a better resolution of thin ligaments.

Scope of the parametric study

In this paragraph, a parametric study of the RCLS method is presented. The parameters considered are the CLS coefficient ϵ and the number of time steps N_s between two re-initialisations of the hyperbolic tangent profile. The relative performance of the method is assessed on the following test cases (defined in Section 4.2.1): Zalesak's slotted disk (128^2 Cartesian mesh) and the disk in a deformation field (256^2 Cartesian mesh). Both of these test cases are performed on Cartesian grids, such that the definitions of ϵ in (7.40) and in (9.37) are equivalent.

In this parametric study, we tested the RCLS with the following parameters:

- $C = 0.5, 1.0$ (see Figure 9.2).
- $N_s = 1, 5, 10$ and no re-initialisation (see Figure 9.3).

Figures 9.2 and 9.3 present the results obtained for:

- Zalesak's slotted disk (middle column), referred to as "Zalesak" in the figures. The contour plots of the scalar transported φ are given at $t = 1s$ (after one circular translation) and the predicted level sets $\varphi = 0.05, 0.5, 0.95$ are shown in black. It is worth recalling that the level set $\varphi = 0.5$ represents the approximate location of the phase boundary and that the inner-most ($\varphi = 0.95$) and outer-most ($\varphi = 0.05$) contour lines gather 90% of the phase transition.

- The disk in a deformation field (right column), referred to as “Spiral” in the figures. The level set contours $\varphi = 0.5$ obtained at $t = 3s$ are drawn in blue.

Choice of ϵ

As expected, when the interface is thickened the method’s ability to capture thin ligaments worsens (see right column in Figure 9.2). However, increasing ϵ stabilises the method (see Section 7.2.4) and smooths the level set contours (see middle column in Figure 9.2).

Also, bearing in mind that the RCLS method is part of a bigger modelling capability, setting the coefficient C to larger values leads to diffused droplets. Although the droplet transfer methodology is designed to cope with such situations, allowing the drops to diffuse unreasonably would result in the over-prediction of coalescence events. In addition, from a physical point of view it is generally desirable to limit the spread of the phase transition as it is the locus of the capillary forces (CSF formulation, see Chapter 3).

From the parametric study, it seems that better results are obtained with $C = 0.5$. Although this value constitutes the limit of stability, the cells of the unstructured computational domain have a low aspect ratio (see Section 9.2.1) and the robustness of the method is maintained with $\epsilon = 0.5\overline{\Delta x}$ for this mesh.

Choice of N_s

Increasing the periodicity of the re-initialisation step improves the ability of the method to capture thin ligaments (see right column in Figure 9.3). This improvement is limited by a maximum value for N_s as the pure advection of φ (no-reinitialisation: $N_s \rightarrow \infty$) is clearly sub-optimal.

Although the largest periodicity tested ($N_s = 10$) provides the best results in terms of ligament length, it leads to convoluted contours of the level set $\varphi = 0.95$ for Zalesak’s slotted disk. Once again, as the surface tension is applied in the region

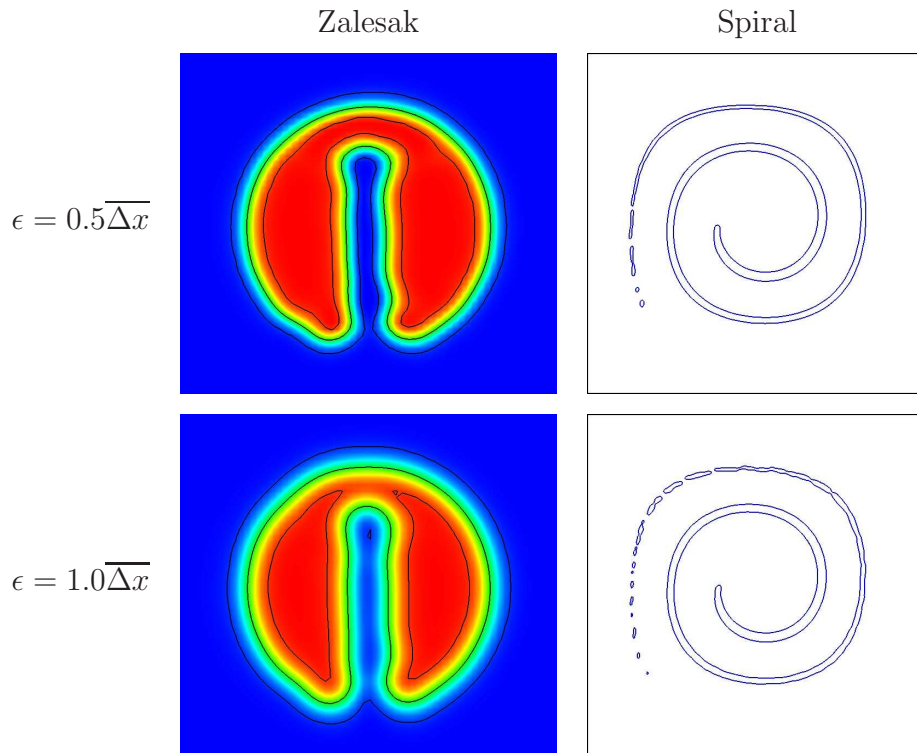


Figure 9.2: Effect of the coefficient ϵ on the performance of the RCLS — Results for $C = 0.5, 1.0$ with $N_s = 5$

of the phase transition, it is desirable to limit the spread of the phase transition and to maintain the regularity of the level set contours.

For this reason, we choose to re-initialise the hyperbolic tangent profile every five time steps: $N_s = 5$. This periodicity gives results similar to the solution performed with $N_s = 1$ at a much lower computational cost ($\sim 25\%$ faster for the translation of the slotted disk).

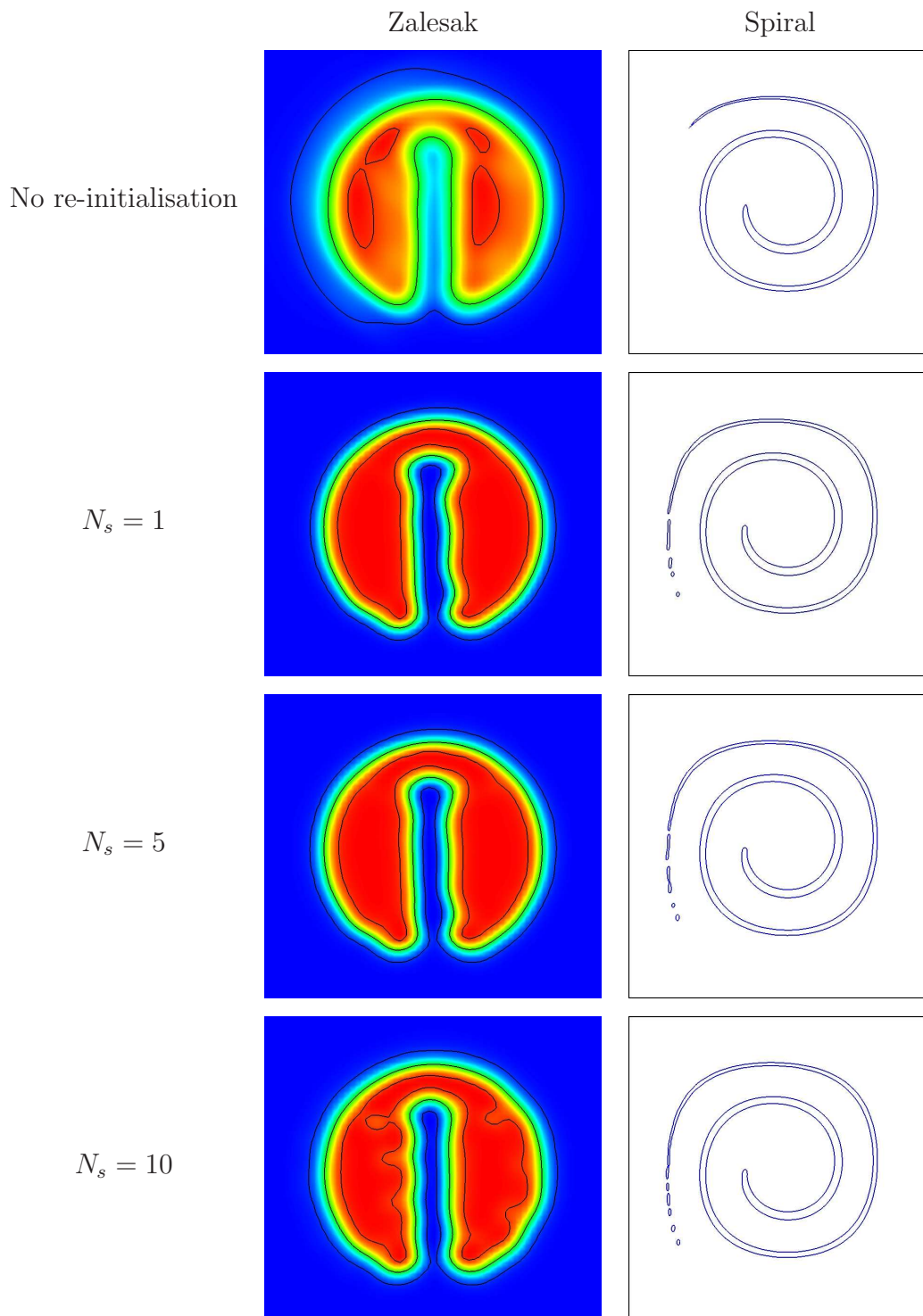


Figure 9.3: Effect of the periodicity of re-initialisation on the performance of the RCLS — Results for $N_s = 1, 5, 10$ and no re-initialisation with $C = 0.5$

Summary

The parameters of the RCLS have been chosen to reach a better trade-off in terms of performance of the RCLS, stability of the method and compliance with the physics of the atomisation. A finer tuning of the method — with a non-uniform value of ϵ in the computational domain — may be performed as a follow-on research project. The chosen set of RCLS parameters is recapitulated in Table 9.3.

Table 9.3: RCLS parameters for the simulation of atomisation

| | |
|---|-------------------------------------|
| Order of the polynomial reconstruction: | $r = 3$ |
| Numerical scheme: | WENO |
| CLS coefficient: | $\epsilon = 0.5\overline{\Delta x}$ |
| Periodicity of the re-initialisation: | $N_s = 5$ |

9.3 Results and discussion

As the emphasis of this numerical study is placed on the mechanisms driving the primary breakup, the computational domain is limited to the close vicinity of the injection plane. Consequently, no droplet characteristics have been extracted from this calculation.

9.3.1 Instabilities of the liquid sheet

The stability analysis performed for a liquid sheet in co-flow [12, 38] demonstrated that the breakup involved undulations of the liquid sheet in sinuous and varicose modes.

In order to illustrate the mechanisms of the sheet breakup, we have extracted the interface contour in both a longitudinal plane \mathcal{P}_l and a transverse plane \mathcal{P}_t of the computational domain (both passing through the centre of the domain, see Figure 9.4). These interface contours are given in Figure 9.5 for \mathcal{P}_l and in Figure 9.6 for \mathcal{P}_t .

Although the interface contours in the longitudinal and transverse planes present different shapes and set of modes, it is clear from Figure 9.5 and Figure 9.6 that the liquid fragmentation is highly non-linear and that the undulations of the sheet involve sinuous and varicose modes in both directions.

As predicted by the theory, our simulation shows — in both the longitudinal and the transverse planes — the initial growth of a sinuous mode: the fundamental of the sheet undulation. Harmonic waves then develop on top of the sinuous oscillation mode of the two surfaces of the sheet. The first harmonic being of varicose type, the growth of the harmonic waves leads to the thinning of the sheet, thus facilitating the breakup of the sheet through aerodynamic tearing (see Figure 9.5).

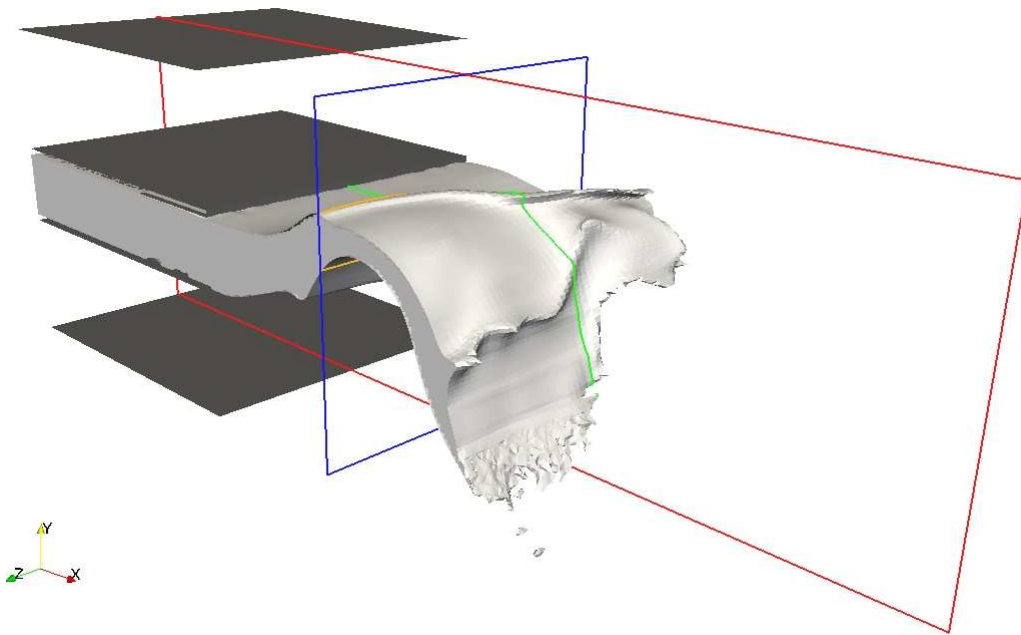


Figure 9.4: Planes of interface contour extraction — In red: the outline of \mathcal{P}_l ; in green: the interface contour in \mathcal{P}_l ; in blue: the outline of \mathcal{P}_t ; in orange: the interface contour in \mathcal{P}_t

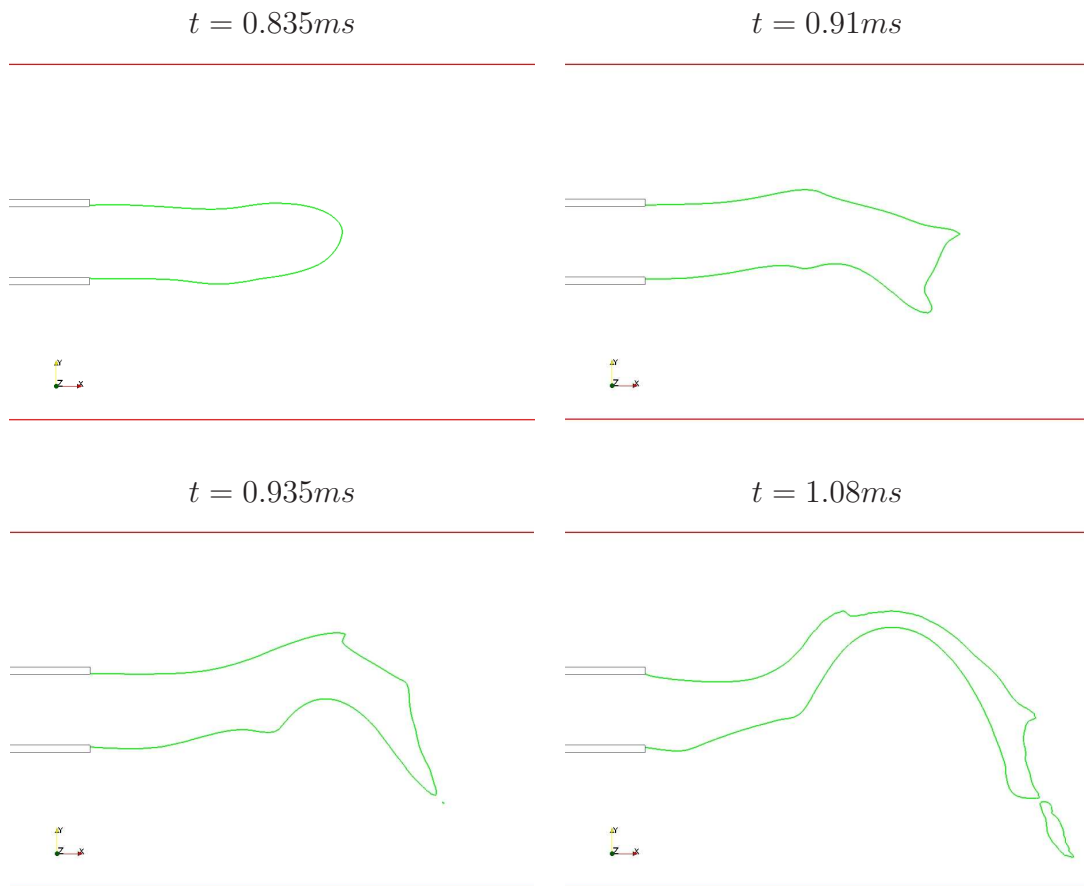


Figure 9.5: Interface contour in the longitudinal plane — Time $t = 0.835ms, 0.91ms, 0.935ms, 1.08ms$

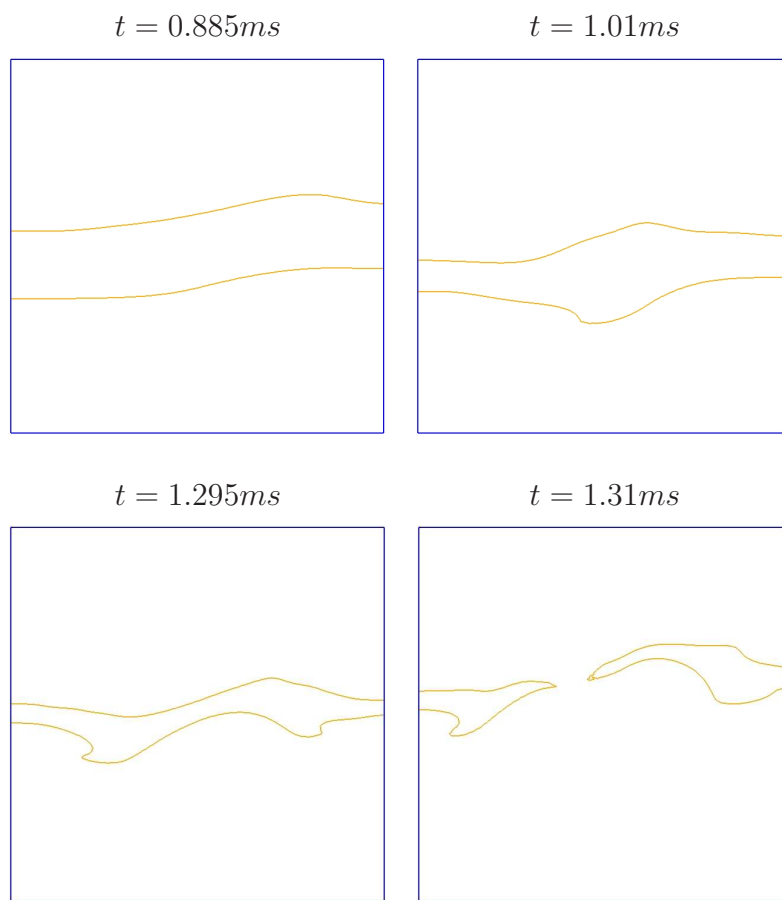


Figure 9.6: Interface contour in the transverse plane — Time $t = 0.885ms, 1.01ms, 1.295ms, 1.31ms$

9.3.2 Torn sheet breakup

Sheet breakup regimes

Fernandez et al. studied the breakup of a flat sheet of water ($300\mu\text{m}$ thick, $U_{liq} \in [1; 2] \text{ ms}^{-1}$) sheared on either side by a stream of gas ($U_{gas} \in [20; 70] \text{ ms}^{-1}$, $p_{gas} \in [1; 6] \text{ bar}$). In their experimental analysis [56], the authors classify the regime of the primary breakup according to the momentum flux ratio M (see (2.2)). In particular, they identify three regimes of liquid sheet breakup:

Up to $M = 0.5$: The cellular breakup. This regime is characterised by the formation of cell-like structures in the liquid sheet through the build-up of longitudinal and transverse undulations of similar wavelength. Such a combination of undulations leads to the bursting of the membranes associated with the cell-like structures and to the creation of spanwise ligaments.

From $M = 0.5$ to $M = 5$ The stretch streamwise ligament breakup. Similarly to the cellular breakup, this regime involves the build-up of both longitudinal and transverse undulations. However, for this regime the fragmentation of the sheet occurs along the longitudinal direction such that the bursting of the membranes is accompanied with the creation of streamwise ligaments.

Above $M = 5$: The torn sheet breakup. As for the previous breakup regime, the torn sheet breakup produces droplets through the disintegration of membranes and via the fragmentation of streamwise ligaments. However, the streamwise ligaments formed have highly irregular shapes and disintegrate through aerodynamic tearing rather than Plateau-Rayleigh instability. This regime also involve the tearing of the continuous region of the liquid sheet into large liquid structures further fragmented by the aerodynamic forces.

In our simulation of the primary breakup, the momentum flux ratio is: $M = 9.5$. According to the above classification, this value of M puts the calculation in the

regime of torn sheet breakup. The flow features predicted by our simulation for this breakup regime are reported in figures 9.7 to 9.11.

Interaction with vortices

The figures of this sub-section present the liquid phase together with an ad hoc iso-surface of the Q-criterion [97] to illustrate the interaction of the turbulence with the liquid sheet. The Q-criterion — defined as the second invariant of the velocity gradient tensor — is widely used to visualise the coherent vortical structures in the flow field [46, 48]. The scalar field Q is given by:

$$Q = \frac{1}{2} ((\text{tr}(\nabla\mathbf{u}))^2 - \text{tr}(\nabla\mathbf{u} \cdot \nabla\mathbf{u})) \quad (9.39)$$

Due to the difference in velocity between the two phases, a shear layer appears on the interface. As a result, a Kelvin-Helmoltz (KH) instability builds up and — as can be seen in figures 9.7 to 9.11 — KH rollers form on either side of the liquid sheet, following initially the shape of the phase interface. As these vortices progress through the computational domain, they break up, thus increasing the level of turbulence downstream of the liquid bulk. In particular, the figures presented illustrate the breakup of vortices through their interaction with the liquid structures.

Evolution of the liquid sheet

The simulation of sheet breakup involves three main phases:

- The build-up of instabilities on the two interface surfaces ($t < 0.935ms$).
- The flapping of the sheet in the longitudinal and transverse directions ($t \in [0.935; 1.25] ms$).
- The tearing of the liquid sheet ($t > 1.25ms$).

Build-up of instabilities This phase starts with the penetration of the liquid in the computational domain and finishes with the first occurrence of the breakup at $t = 0.935ms$. It involves the initial deformation of the interface in the injection channel and the development of sinuous and varicose modes of undulation of the liquid sheet (see Figure 9.7). As described in Section 9.3.1, the growth of surface waves — in both the longitudinal and the transverse directions — leads to the thinning of the liquid sheet, thus facilitating the first pinch-off event by aerodynamic tearing.

Sheet flapping This phase involves the flapping of the sheet in both the longitudinal and the transverse directions, similarly to a flag. For this stage, the simulation predicts the disintegration of the membranes formed at the peaks and the troughs of the undulations (see Figure 9.8). The membrane puncturing is accompanied by the formation and the pinch-off of streamwise ligaments (see Figure 9.9). These flow features suggest that the breakup regime is the stretch streamwise ligament breakup. It constitutes a transition before the establishment of the torn sheet breakup.

Sheet tearing From $t = 1.25ms$, the computation predicts the tearing of the sheet in the transverse and then the longitudinal directions (see Figure 9.10 and Figure 9.11 respectively).

At $t = 1.25ms$, the liquid sheet presents a bag-like structure bent towards the top of the domain. As the sheet significantly obstructs the gas stream, it is subjected to relatively high aerodynamic forces and a tear is initiated. While the tear propagates in the transverse direction — following the path of minimum sheet thickness — the liquid structure being torn away undergoes a membrane-type breakup.

Then, at time $t = 1.3ms$, a bag-like structure is formed in the centre of the sheet. This structure is punctured and the aerodynamic forces initiate a longitudinal tear in the hole produced. As the tear propagates upstream in the liquid phase, two streamwise ligaments are formed. These ligaments develop bag-like structures

themselves and get subsequently torn away by the gas stream.

The flow field predicted by the numerical simulation in the “sheet tearing” phase, matches closely the description of the torn sheet breakup given in [56]. This validates qualitatively our simulation of the liquid sheet breakup.

Our computation of the liquid sheet breakup highlights the potential of the numerical approach to study the mechanisms of the primary breakup. Indeed, the simulation described the entire breakup process while the stability analysis could only predict its onset and experiments could only capture the sheet tearing. In addition, only the CFD can provide insight on the interaction of the vortices in the gas phase with the liquid structures.

9.3.3 Breakup length

As a first step towards a quantitative validation of this numerical simulation of sheet breakup, the average breakup length predicted by the computation is compared to the value given by the correlation provided in [56].

In [56], Fernandez et al. relate the breakup length L_b to a non-dimensional number. Recalling that h is the height of the fuel injection channel and noting We_h , the Weber number based on h , this relation reads:

$$\frac{L_b}{h} = f \left(\sqrt[3]{\frac{U_{liq}}{U_{gas}} \frac{1}{MW e_h}} \right) \quad (9.40)$$

For the conditions of the calculation, the non-dimensional number is equal to 0.008, which leads to $\frac{L_b}{h} = 6 \pm 2$. In our numerical experiment, we observe an average breakup length of $\overline{\frac{L_b}{h}} = 5$ which is well within the range predicted by the correlation.

It is also worth noting that (9.40) is based on a restricted range of flow conditions (water liquid sheet ; air pressure below 6bar), and that it accounts for a limited range

of parameters (ignores Re_{liq}). A correlation based on a more exhaustive experimental test campaign may further confirm the prediction of our simulation.

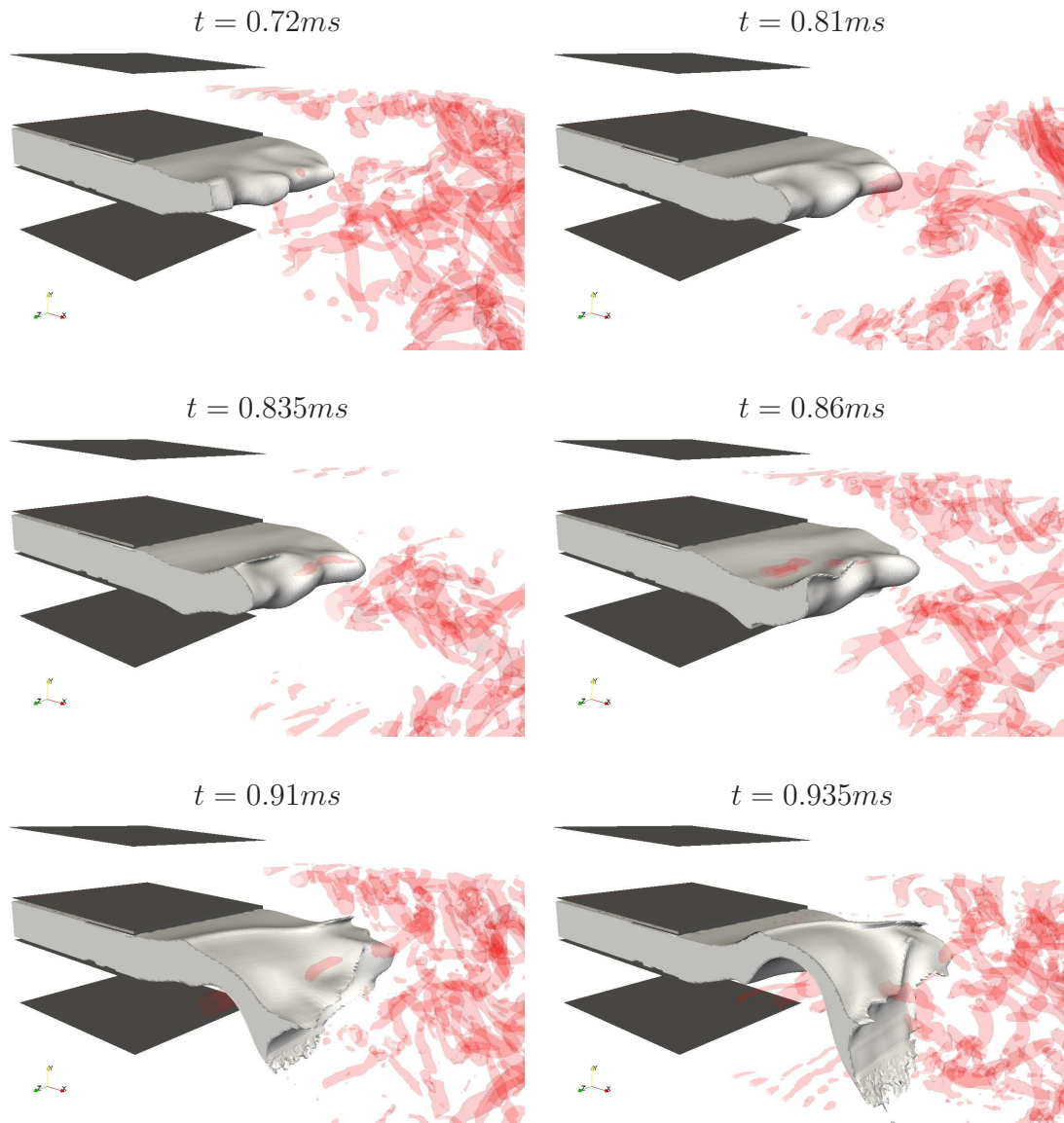


Figure 9.7: Simulation of flat sheet breakup with *lesRCLSfoam* — Liquid phase (in grey) and an ad hoc iso-surface of Q-criterion (in transparent red) for the build-up of instabilities

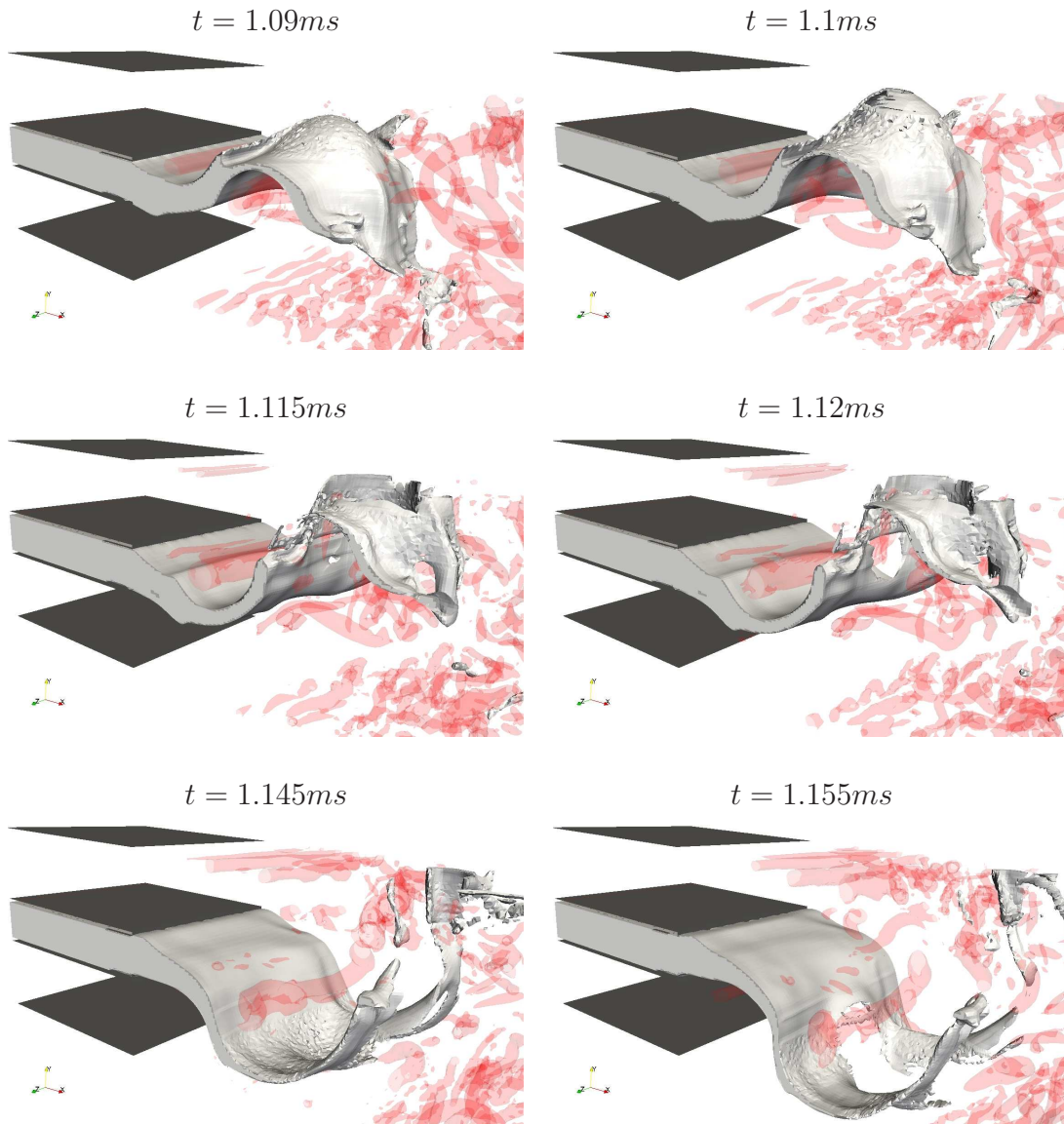


Figure 9.8: Simulation of flat sheet breakup with *lesRCLSFoam* — Liquid phase (in grey) and an ad hoc iso-surface of Q-criterion (in transparent red) for the flapping, the membrane puncturing and the ligament formation

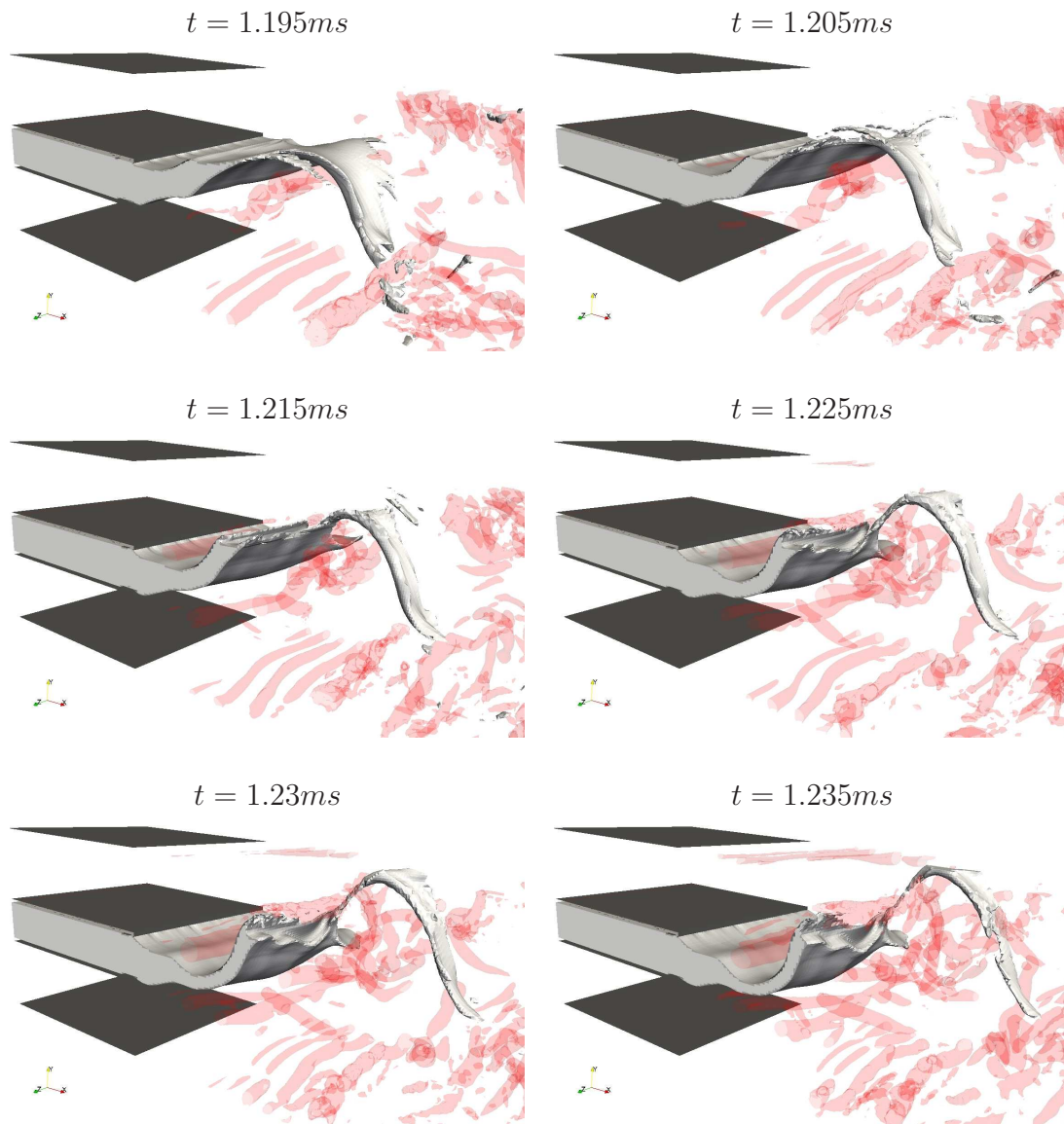


Figure 9.9: Simulation of flat sheet breakup with *lesRCLSFoam* — Liquid phase (in grey) and an ad hoc iso-surface of Q-criterion (in transparent red) for the pinch-off of a streamwise ligament

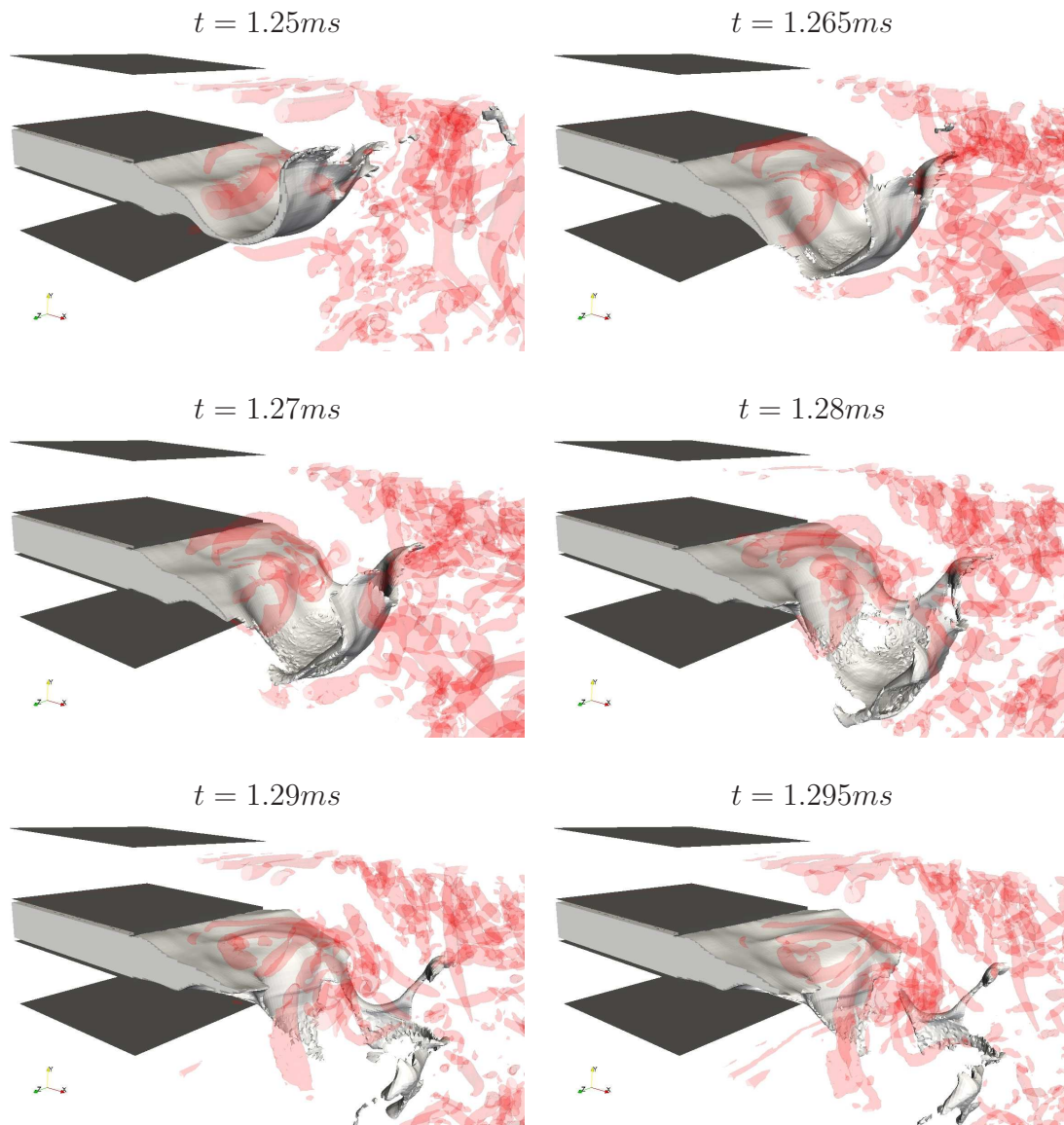


Figure 9.10: Simulation of flat sheet breakup with *lesRCLSFoam* — Liquid phase (in grey) and an ad hoc iso-surface of Q-criterion (in transparent red) for the tearing of the sheet in the transverse direction

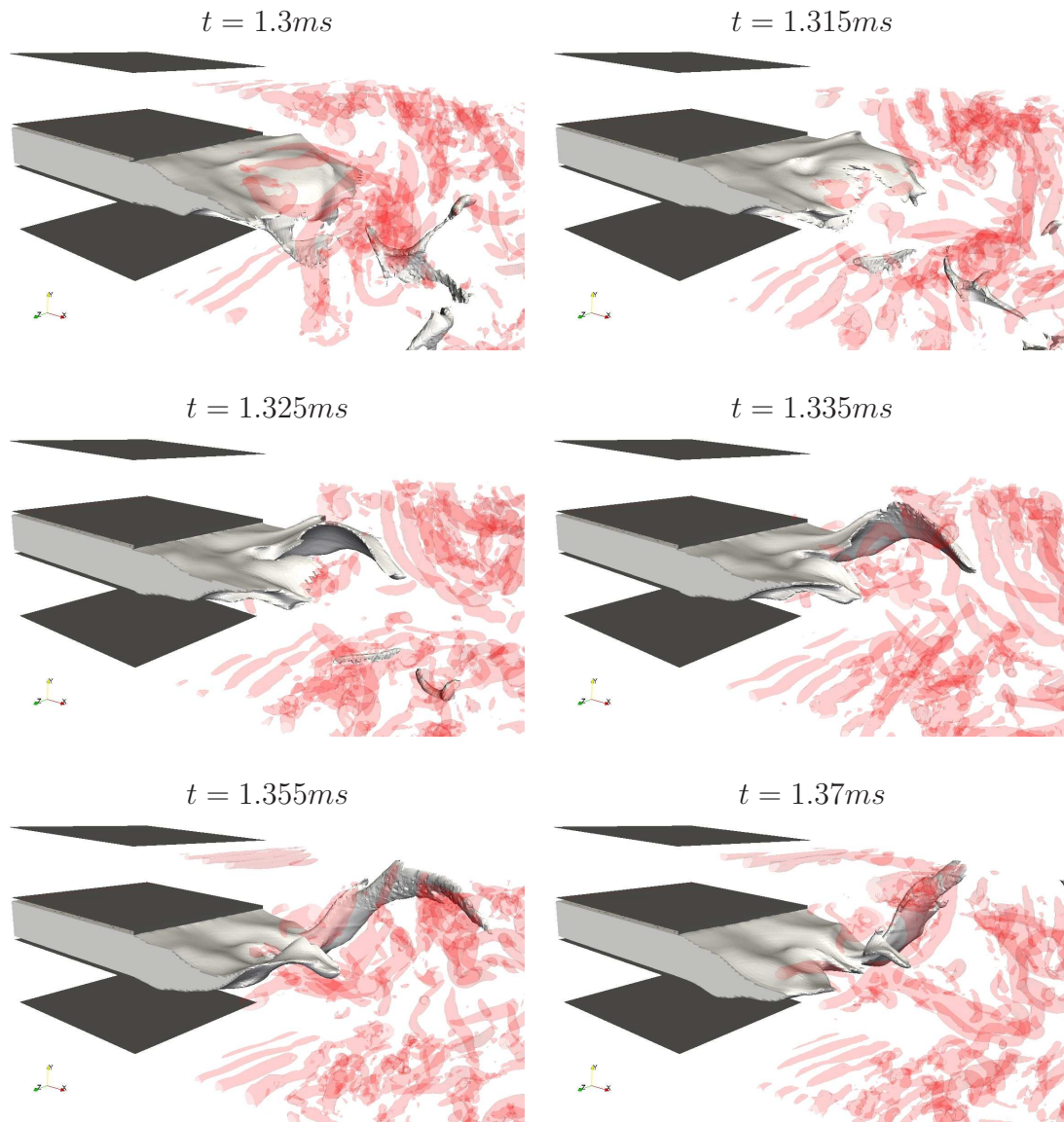


Figure 9.11: Simulation of flat sheet breakup with *lesRCLSFoam* — Liquid phase (in grey) and an ad hoc iso-surface of Q-criterion (in transparent red) for the tearing of the sheet in the longitudinal direction

Chapter 10

Conclusion

The main accomplishment of this work consists in the generation of a modelling capability for the parallel computation of the fuel injection in aero-engine combustion chambers. To reach this goal, a modelling strategy has been produced based on the existing methodologies in the field of multiphase CFD and the requirements of the aeronautical industry. This set of guidelines lead to the creation of a novel WENO scheme for general polyhedral meshes, the development of a robust conservative level set method to transport the interface and the generation of an algorithm to output droplet boundary conditions to combustion codes. In this chapter the main achievements of this research project are summarised in Section 10.1 and some follow-on research topics are suggested in Section 10.2.

10.1 Achievements

10.1.1 Modelling strategy to simulate fuel injection

A comprehensive literature review has been produced, reporting the current physical description of the phenomenon and a broad overview of the numerical methods available to simulate multiphase flows. In particular, the existing numerical frameworks, the various interface description methods and the means of handling the singulari-

ties introduced by the interface are described in detail. Finally, the state-of-the-art methods developed by scientists to simulate the atomisation process are presented and the current trends in atomisation modelling are outlined.

Also, two open source multiphase codes have been identified: OpenFOAM and Gerris. Their two-phase flow modelling capabilities have been thoroughly tested against typical performance tests and both codes have been demonstrated on the simulation of primary breakup. Although Gerris performed remarkably well, some non-physical behaviours appeared when tested on more advanced configurations (see Section 4.3.2), suggesting that the modelling tool is not yet mature. By virtue of its design (DNS, not developer-friendly, ongoing development), Gerris is aimed at academic research rather than component design. OpenFOAM's two-phase flow capability performed rather poorly on typical numerical tests (low-order schemes, non-physical wiggles on the interface). However, this code appeared to be a good platform to implement a multiphase CFD tool for the industry (unstructured approach, flexible framework).

Due to the lack of appropriate modelling capability we decided to develop a novel numerical tool to simulate the atomisation process. We chose to use the OpenFOAM "C++ toolbox" as the code vehicle for this CFD solver.

Based on the literature review and considering the requirements of the aeronautical industry, a modelling strategy has been identified to simulate the fuel injection in real engine combustion chambers. In particular, this methodology suggested the development of:

- A numerical tool performing on unstructured meshes.
- An efficient interface description technique that conserves mass while predicting accurately the location of the interface.
- An algorithm to output droplet boundary conditions for combustion codes.

The conservative level set method of Olsson et al. [163, 164] has been identified as

the interface description technique that provides the best trade-off between accuracy and computational cost. In order to extend this method to problems involving large density ratios, we chose to transport the level set scalar using WENO schemes. As no WENO schemes existed for general unstructured meshes, we developed a methodology — implemented in parallel — for the construction of a high-order WENO schemes on general polyhedral unstructured meshes.

10.1.2 Novel WENO scheme for unstructured meshes

Our method improves and extends the approach of Dumbser and Käser [49] — generated for tetrahedral meshes — to polyhedral meshes through a more general derivation of the reconstruction operator and the inter-cell fluxes. In addition, we have handled efficiently the notorious complexity of high-order schemes on 3D mixed-element grids by generating novel algorithms.

Principally, these algorithms include the tetrahedralisation of the mesh, which allows generality of the approach while remaining efficient and affordable, together with a novel approach to stencil generation and a faster interpolation of the solution. The general method for tetrahedralisation of the mesh is presented for convex polyhedral cells with convex polygonal faces. Also, we have ensured that as much as possible of the computational work is done in pre-processing steps, in order to reduce the work done at run time.

Finally, the derivation of the resulting inter-cell fluxes is given in the case of convex polyhedral cells for linear hyperbolic systems of equations. The application of the method to the level set equation is also given, with an interpretation of the Riemann problem in such frameworks. The performance of the scheme presented has been demonstrated on typical two-dimensional and three-dimensional test cases of the level set method. The results obtained with the WENO3 scheme without re-distancing compare very well with existing methods that use WENO schemes of order five together with re-distancing (see [37, 246]). Besides, the numerical conver-

gence studies conducted on various types of mesh and the extension of the method to the solution of a non-linear hyperbolic PDE have demonstrated the expected performance of the scheme.

10.1.3 Mass-conservative interface description

The conservative level set (CLS) method of Olsson et al. [163, 164] was chosen as starting point for our novel interface description technique because this technique:

- Conserves mass well at a lower cost than hybrid methods [71].
- Resolves the interface accurately as it is based on a level set formulation.
- Has been demonstrated on the atomisation of Diesel jet (simple configuration) by Desjardins et al. [47].

However, the conservative level set method needed to be improved in terms of stability and accuracy. Besides as the CLS was developed for Cartesian grids, the method had to be extended to unstructured grids.

The extension to general polyhedral meshes and the improvement in terms of stability were obtained by solving the advection equation and the re-initialisation equations of the CLS method with the WENO scheme presented in Chapter 6. In particular, this numerical scheme significantly improved the stability of the method in the presence of very large density ratios as demonstrated by the simulation of the falling droplet (see Section 8.4.2). For this calculation, our modelling capability remained stable regardless of the numerics employed to solve the pressure-velocity coupling whereas OpenFOAM's original solver crashed when tested with advanced numerical methods (Crank-Nicholson time scheme, GAMG solver).

The improvement of the accuracy of the CLS method originated from two sources: the high-order accurate numerical scheme mentioned above and the addition of a flux-limiter algorithm to the transport of the level set scalar. This additional step insures that the solution remains bounded.

Whereas the conservative level set methods of Olsson et al. and Desjardins et al. treat the scalar field as a level set, we choose to consider it as a smeared out liquid volume fraction. The addition of the flux limiter, specific to VOF methods, extends the CLS further towards a volume of fluid method and guarantees that the liquid volume fraction remains physical everywhere in the computational domain ($0 \leq \varphi \leq 1$).

In addition, taking advantage of the polynomial reconstruction performed by the WENO scheme, we have created a novel method to produce a high-order accurate calculation of the gradient of the level set field. This gradient is required for the calculation of the curvature and the surface tension forces.

The resulting method named robust conservative level set (see Chapter 7) conserves mass to machine accuracy (see Section 8.4.1) and outperforms the ACLS method of Desjardins et al. (see Section 7.5). Besides, numerical testing of the RCLS has demonstrated its ability to capture the physics accurately (see Section 8.4).

10.1.4 Modelling capability for the simulation of atomisation

In order to be readily applicable to the simulation of atomisation, the novel interface description technique needed to be embedded in a bigger framework:

- Allowing for the solution of the pressure and velocity fields.
- Outputting droplet characteristics for the simulation of the reacting flow in the whole combustion chamber.
- Parallelised to make the most of the available computational power.

As the main focus of the work has been placed on the transport of the liquid volume fraction, the modelling capability has adopted OpenFOAM's methodology for the solution of the incompressible Navier-Stokes equations and its coupling with the transport of the liquid volume fraction (see Section 8.1).

The algorithm that outputs the droplets boundary conditions is made of a set of routines that selects the droplets in the level set field, calculates their characteristics and writes them out into data files. This method has been implemented and demonstrated in parallel (see Section 8.2).

A significant amount of effort has been placed on the development of a n-halo parallelisation of the modelling capability (see Section 8.3). Indeed, OpenFOAM comes with a 0-halo approach that is only sufficient in the case of low-order numerics and a methodology has been coded to allow for the high-order scheme produced to operate at the expected order of accuracy in a parallelised computational domain.

10.1.5 Demonstration of the numerical tool on the primary breakup

The modelling capability has been demonstrated on the simulation of the primary breakup of a liquid sheet in a co-flowing stream of gas. The calculation was performed using a quasi-DNS/LES methodology [71] (see Section 9.1) and the problem was set with material properties and boundary conditions relevant to the injection of fuel in an aero-engine combustor (see Section 9.2).

The computation has demonstrated the ability of the modelling tool to capture the physics accurately (see Section 9.3) and further illustrates the potential of the numerical approach. The numerical results have been validated qualitatively against theoretical predictions (stability analysis) [12, 38] and experimental data [56]. In particular, the modes of undulations of the liquid sheet were correctly captured by the code and the expected breakup regime was observed in the simulation results.

Finally, the computation reproduced faithfully the breakup length predicted by a correlation based on experimental data [56]. This constitutes a first step towards a quantitative validation.

10.2 Follow-on research topics

10.2.1 Application to aeronautical fuel-injectors

The validation of the modelling tool against experimental data is the next logical step. This validation exercise should be performed on two kinds of geometries:

- Idealised experimental set-ups.
- Real engine configurations.

Even on simplified geometries tested in a well controlled and monitored environment, the primary breakup remains very difficult to predict quantitatively. The first step towards the validation of the numerical capability should therefore focus on reproducing flat liquid-sheet disintegration experiments [56, 135, 140, 141].

As the work of Fernandez et al. [56] is particularly relevant to the aeronautical application and provides both qualitative (breakup regimes) and quantitative (global oscillation frequencies, transverse wavelengths and breakup lengths) results, these experimental data should be compared against CFD results. The tuning of the interface description method should be performed in parallel to this task.

Once validated on idealised configurations, the method should be calibrated against experimental data obtained for real fuel-injector geometries.

For both idealised and realistic geometries, the validation against droplet size distributions and droplet velocity distributions would provide the most useful assessment of the performance of the numerical tool.

10.2.2 Improvement of the interface description technique

Although promising, the interface description technique — as described in this thesis — may be improved by the implementation of additional numerical techniques. In particular, we suggest the following developments for our method:

- Implementation of a non-uniform CLS coefficient ϵ for non-Cartesian computational domains. Special care would have to be taken for the extension to parallel computations in order to maintain the mass conservation.
- Development of a ghost fluid method for unstructured grids. Such treatment of the singularities in the flow field is likely to improve the accuracy of the numerical method. However, this approach — followed by Desjardins et al. on Cartesian grids [46, 47] — may lead to an unstable CFD solver [2].
- Implementation of the arbitrary high-order schemes using derivatives (ADER) of Toro et al. [110, 221, 222, 252, 253, 256–258]. This approach — based on the concept of arbitrary high-order generalised Riemann solvers — reaches high-order accuracy in both time and space in a single step. Adding the ADER method to the numerical tool is expected to improve the time accuracy while reducing the computational cost as the WENO reconstruction would only be performed once per time step (instead of 3 times per time step with the Runge-Kutta scheme currently implemented).
- Correction of the parallel implementation of OpenFOAM’s solvers for algebraic equations. Due to the 0-halo approach of the code, the solvers coded (GAMG and PCG) lead to a small mass error in parallel (see Section 8.4). Correcting this problem of OpenFOAM would result in computations of two-phase flows that conserve mass to machine accuracy in parallel.
- Implementation of a Lagrange droplet tracking scheme (with secondary breakup modelling) to simulate the whole atomisation process with a single CFD solver.

10.2.3 Extension of the modelling capability

As coded, the method is applicable to incompressible flows with no phase change. Although such framework is generally adopted by scientists to study the primary

breakup in combustion chambers, the numerical tool could be extended to embed more physics. In particular, we suggest extending the modelling capability to:

- Compressible flows. In such instance, the velocity field would no longer be solenoidal ($\nabla \mathbf{u} \neq 0$).
- Vaporisation modelling. This physical process is generally handled by adding a term in the level set equation [247].

In such instances, it is preferable to consider the level set equation as a Hamilton-Jacobi equation. Therefore, the numerical scheme presented in Chapter 6 should be extended to the solution of Hamilton-Jacobi equations. A significant amount of work has already been produced for the solution of Hamilton-Jacobi equations with WENO schemes [5, 11, 14, 94, 106, 171, 285]. Combining the work of Hu and Shu [94] with the methodology employed in Section 7.3 to produce a high-order calculation of the gradient would be a good starting point for this project.

10.2.4 Development of sub-grid scale models

As mentioned in chapters 3 and 5, the accurate modelling of the atomisation process with a LES framework requires the development of sub-grid scale models. These models should concern:

- The sub-grid contributions of the interface to the flow field. These have been categorised by Labourasse et al. in [118]. As the turbulence is believed to play a strong role on the onset of the breakup [54, 146], the creation of SGS models for the turbulence-interface interaction should be a priority.
- The sub-grid contributions of the smallest liquid structures to the flow field. This research project could start from the SGS models used in the Euler/Euler LES approach [67, 200].

10.2.5 Super-critical fuel injection

As the current trends in the design of combustion chambers lead to the ever increasing pressure of the gas, the fuel injection is more and more likely to occur at super-critical conditions. Under these conditions, the surface tension tends to zero and the fluids become miscible.

The numerical simulation of this type of injection is becoming a priority for aero-engine manufacturers as it is crucial to the prediction of combustors performance at high power settings. As it involves very specific issues, this problem cannot simply be considered as an extension of the multiphase flow case and thus requires its own research programme.

Bibliography

- [1] Private communication with Berlemont — CORIA.
- [2] Private communication with Soteriou — United Technologies Research Center.
- [3] Private communication with Toro — University of Trento.
- [4] R. Abgrall. On Essentially Non-Oscillatory schemes on unstructured meshes: analysis and implementation. *Journal of Computational Physics*, 144:45–58, 1994.
- [5] R. Abgrall. Numerical discretization of the first-order Hamilton-Jacobi equation on triangular meshes. *Communications on Pure and Applied Mathematics*, 49:1339–1373, 1998.
- [6] S. Apte, M. Gorokhovski, and P. Moin. LES of atomizing spray with stochastic modeling of secondary break-up. *International Journal of Multiphase Flow*, 29:1503–1522, 2003.
- [7] S. Apte, K. Mahesh, P. Moin, and J. Oefelein. LES of swirling particle-laden flows in a coaxial-jet combustor. *International Journal of Multiphase Flow*, 29:1311–1331, 2003.
- [8] T. Arai and H. Hashimoto. Disintegration of a thin liquid sheet in a co-current gas stream. *Proceedings of International Conference on Liquid Atomization and Spray Systems*, 1985.

- [9] N. Ashgriz and J. Poo. FLAIR, flux line-segment model for advection and interface reconstruction. *Journal of Computational Physics*, 93:449–468, 1991.
- [10] E. Aulisa, A. Manservigi, R. Scardovelli, and S. Zaleski. Volume of Fluid method for the dynamics of free boundaries. *Journal of Computational Physics*, 192:355–364, 2003.
- [11] M. Bardi and S. Osher. The non-convex multi-dimensional Riemann problem for Hamilton-Jacobi equations. *SIAM Journal on Mathematical Analysis*, 22:344–351, 1990.
- [12] F. Barreras. *Experimental study of the break-up and atomization of a planar liquid sheet*. PhD thesis, University of Zaragoza, 1998.
- [13] T. Barth and P. Frederickson. High order solution of the Euler equations on unstructured grids using quadratic reconstruction. Technical Report 90-0013, American Institute of Aeronautics and Astronautics, 1990.
- [14] T. Barth and J. Sethian. Numerical schemes for the Hamilton-Jacobi and Level Set equations on triangulated domains. *Journal of Computational Physics*, 145:1–40, 1998.
- [15] J. Beale and R. Reitz. Modeling spray atomization with the Kelvin-Helmholtz/Rayleigh-Taylor hybrid model. *Atomization and Sprays*, 9:623–650, 1999.
- [16] J. Bell, C. Dawson, and G. Shubin. An unsplit, higher order Godunov method for scalar conservation laws in multiple dimensions. *Journal of Computational Physics*, 74:1–24, 1988.
- [17] R. Benzi, S. Succi, and M. Vergassolac. The lattice Boltzmann equation: Theory and Applications. *Physics Reports*, 222:145–197, 1992.

- [18] G. Bianchi, F. Minelli, R. Scardovelli, and S. Zaleski. 3D large scale simulation of the high-speed liquid jet atomization. Technical Report 2007-01-0244, Society of Automotive Engineers, 2007.
- [19] G. Bianchi, P. Pelloni, S. Toninel, R. Scardovelli, A. Leboissetier, and S. Zaleski. Improving the knowledge of high-speed liquid jets atomization by using quasi-direct 3D simulation. Technical Report 2005-24-089, Society of Automotive Engineers, 2005.
- [20] S. Billett and E. Toro. *Numerical methods for wave propagation*, chapter Unsplit WAF-type schemes for three-dimensional hyperbolic conservation laws, pages 75–124. Kluwer Academic Publishers, 1998.
- [21] M. Boivin, O. Simonin, and K. Squires. On the prediction of gas-solid flows with two-way coupling using large eddy simulation. *Physics of Fluids*, 12:2080–2090, 2000.
- [22] D. Book, J. Boris, and K. Hain. Flux-Corrected Transport II: Generalization of the method. *Journal of Computational Physics*, 18:248–283, 1975.
- [23] J. Boris and D. Book. Flux-Corrected Transport I: SHASTA, a fluid transport algorithm that works. *Journal of Computational Physics*, 11:38–69, 1973.
- [24] J. Boris and D. Book. Flux-Corrected Transport III: Minimal-error FCT algorithms. *Journal of Computational Physics*, 20:397–431, 1976.
- [25] A. Bourlioux. A coupled Level Set and Volume of Fluid algorithm for tracking material interfaces. *Proceedings of the 6th International Symposium On Computational Fluid Dynamics*, 6:15–22, 1995.
- [26] J. Brackbill, D. Kothe, and C. Zemach. A continuum method for modeling surface tension. *Journal of Computational Physics*, 100:335–354, 1992.

- [27] N. Bremond. *Stabilité et atomisation des nappes liquides*. PhD thesis, Université de Provence - Aix-Marseille I - IRPHE, 2003.
- [28] M. Brocchini and D. Peregrine. The dynamics of strong turbulence at free surfaces. part I Description. *Journal of Fluid Mechanics*, 449:225–254, 2001.
- [29] M. Brocchini and D. Peregrine. The dynamics of strong turbulence at free surfaces. part II Free-surface boundary conditions. *Journal of Fluid Mechanics*, 449:255–290, 2001.
- [30] J. Cahn and J. Hilliard. Free energy of a nonuniform system. part III Nucleation in a two-component incompressible fluid. *Journal of Chemical Physics*, 31:688–699, 1959.
- [31] H. Carentz. *Etude de la pulvérisation d'une nappe liquide mince*. PhD thesis, Université Pierre et Marie Curie - Paris 6, 2000.
- [32] Y. Chang, T. Hou, B. Merriman, and S. Osher. A Level-Set formulation of Eulerian capturing methods for incompressible fluid flows. *Journal of Computational Physics*, 124:449–464, 1996.
- [33] B. Chehroudi, Y. Onuma, S.-H. Chen, and F. Bracco. On the intact core of full-cone sprays. Technical Paper 850126, Society of Automotive Engineers, 1985.
- [34] D. Chopp. *Computation of minimal surfaces via mean-curvature flow*. PhD thesis, University of California Berkeley, 1991.
- [35] A. Chorin. Flame advection and propagation algorithms. *Journal of Computational Physics*, 35:1–11, 1980.
- [36] P. Colella. Multidimensional upwind methods for hyperbolic conservation laws. *Journal of Computational Physics*, 87:171–200, 1990.

- [37] F. Couderc. *Développement d'un code de calcul pour la simulation d'écoulements de fluides non miscibles. Application à la désintégration assistée d'un jet liquide par un courant gazeux*. PhD thesis, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace — Toulouse, 2007.
- [38] J. Cousin and C. Dumouchel. Effect of viscosity on linear instability of a flat liquid sheet. *Atomization and Sprays*, 6:563–576, 1996.
- [39] G. Crapper, N. Dombrowski, and G. Pyott. Large amplitude Kelvin-Helmholtz waves on thin liquid sheets. *Proceedings of the Royal Society of London, Series A*, 342:209–224, 1975.
- [40] B. Daly. Numerical study of two fluid Rayleigh-Taylor instability. *Physics of Fluids*, 10:297–307, 1967.
- [41] L. Davidson. Large eddy simulations: A note on derivation of the equations for the subgrid turbulent kinetic energies. Technical Report 97/12, 980904, Chalmers University of Technology, Gothenburg, Sweden, 1997.
- [42] P.-G. de Gennes, F. Brochart-Wyart, and D. Quere. *Capillarity and Wetting Phenomena: Drops, Bubbles, Pearls, Waves*. Berlin: Springer, 2003.
- [43] E. de Villiers. *The potential of large eddy simulation for the modelling of wall bounded flows*. PhD thesis, Imperial College of Science, Technology & Medicine, 2006.
- [44] E. de Villiers, A. Gosman, and H. Weller. Large eddy simulation of primary Diesel spray atomization. Technical Report 2004-01-0100, Society of Automotive Engineers, 2004.
- [45] R. DeBar. A method in 2D Eulerian hydrodynamics. Technical Report UCID-19683, Lawrence Livermore National Laboratory, 1974.

- [46] O. Desjardins. *Numerical methods for liquid atomization and application in detailed simulations of a diesel jet*. PhD thesis, Stanford University, 2008.
- [47] O. Desjardins, V. Moureau, E. Knudsen, M. Herrmann, and H. Pitsch. Conservative Level Set/ghost fluid method for simulating primary atomization. *Proceedings of the 20th Annual Conference of the Institute for Liquid Atomization and Spray Systems - Americas*, 2007.
- [48] Y. Dubief and F. Delcayre. On coherent-vortex identification in turbulence. *Journal of Turbulence*, 1:1–22, 2000.
- [49] M. Dumbser and M. Käser. Arbitrary high order non-oscillatory finite volume schemes on unstructured meshes for linear hyperbolic systems. *Journal of Computational Physics*, 221:693–723, 2007.
- [50] M. Dumbser, M. Käser, V. Titarev, and E. Toro. Quadrature-free non-oscillatory finite volume schemes on unstructured meshes for nonlinear hyperbolic systems. *Journal of Computational Physics*, 226:204–243, 2007.
- [51] C. Duquennoy, O. Lebaigue, and J. Magnaudet. Physical and numerical modelling of a gas-liquid-solid contact line. *Proceedings of the 37th European Two-Phase Flow Group Meeting*, 1999.
- [52] J. Eggers and E. Villermaux. Physics of liquid jets. *Reports on Progress in Physics*, (71:036601), 2008.
- [53] D. Enright, R. Fedkiw, J. Ferziger, and I. Mitchell. A hybrid Particle Level Set method for improved interface capturing. *Journal of Computational Physics*, 183:83–116, 2002.
- [54] G. Faeth, L.-P. Hsiang, and P.-K. Wu. Surface and breakup properties of sprays. *International Journal of Multiphase Flow*, 21:99–127, 1995.

- [55] R. Fedkiw, T. Aslam, B. Merriman, and S. Osher. A non-oscillatory eulerian approach to interfaces in multimaterial flows (the Ghost Fluid Method). *Journal of Computational Physics*, 152:457–492, 1999.
- [56] V. Fernandez, P. Berthoumie, and Lavergne G. Liquid sheet disintegration at high pressure: An experimental approach. *Comptes Rendus Mécanique*, 337:481–491, 2009.
- [57] G. Forsythe, M. Malcolm, and C. Moler. *Computer methods for mathematical computations*. NJ: Prentice-Hall, 1977.
- [58] G. Forsythe and C. Moler. *Computer solution of linear algebraic systems*. NJ: Prentice-Hall, 1967.
- [59] M. François, S. Cummins, E. Dendy, D. Kothe, J. Sicilian, and M. Williams. A balanced-force algorithm for continuous and sharp interfacial surface tension models within a volume tracking framework. *Journal of Computational Physics*, 213:141–173, 2006.
- [60] O. Friedrich. Weighted Essentially Non-Oscillatory schemes for the interpolation of mean values on unstructured grids. *Journal of Computational Physics*, 144:194–212, 1998.
- [61] M. Fritts, W. Cowley, and H. Trease. The free Lagrange method. *Lecture Notes on Physics, New York: Springer-Verlag*, 238, 1985.
- [62] M. Fulgosi, D. Lakehal, S. Banerjee, and V. Angelis. Direct numerical simulation of turbulence in a sheared air-water flow with a deformable interface. *Journal of Fluid Mechanics*, 482:319–345, 2003.
- [63] C. Fureby, A. Gosman, G. Tabor, and H. Weller. Large eddy simulation of turbulent channel flows. *Turbulent shear flows*, 11, 1997.

- [64] C. Fureby, G. Tabor, H. Weller, and A. Gosman. A comparative study of subgrid scale models in homogeneous isotropic turbulence. *Physics of Fluids*, 9:1416–1429, 1997.
- [65] D. Fuster, A. Bague, T. Boeck, L. Le Moyne, A. Leboissetier, S. Popinet, P. Ray, R. Scardovelli, and S. Zaleski. Simulation of primary atomization with an octree adaptive mesh refinement and VOF method. *International Journal of Multiphase Flow*, 35:550–565, 2009.
- [66] D. Fyfe, E. Oran, and M. Fritts. Surface tension and viscosity with Lagrangian hydrodynamics on a triangular mesh. *Journal of Computational Physics*, 76:349–384, 1988.
- [67] M. Garcia, E. Riber, O. Simonin, and T. Poinso. Comparison between Euler/Euler and Euler/Lagrange LES approaches for confined bluff-body gas-solid flow prediction. *Proceedings of the 6th International Conference on Multiphase Flows*, 2007.
- [68] R. Gingold and J. Monaghan. Smoothed particle hydrodynamics: Theory and Application to non-spherical stars. *Royal Astronomical Society, Monthly Notices*, 181:375–389, 1977.
- [69] J. Glimm, O. McBryan, R. Menikoff, and D. Sharp. Front tracking applied to Rayleigh-Taylor instability. *SIAM Journal on Scientific and Statistical Computing*, 7:230–251, 1987.
- [70] S. Godunov. A finite difference method for the computation of discontinuous solutions of the equations of fluid dynamics. *Matematicheskii Sbornik*, 47:271–306, 1959.
- [71] M. Gorokhovski and M. Herrmann. Modeling primary atomization. *Annual Review of Fluid Mechanics 2008*, 40:343–366, 2008.

- [72] F. Grinstein, L. Margolin, and W. Rider. *Implicit Large Eddy Simulation*. Cambridge University Press, 2007.
- [73] D. Gueyffier, J. Li, A. Nadin, R. Scardovelli, and S. Zaleski. A Volume of Fluid interface tracking with smoothed surface stress methods for three-dimensional flows. *Journal of Computational Physics*, 152:423–456, 1999.
- [74] A. Gunstensen. *Lattice-Boltzmann studies of multiphase flow through porous media*. PhD thesis, MIT, Cambridge, MA, 1992.
- [75] W. Hagerty and J. Shea. A study of the stability of plane fluid sheets. *Transactions of ASME: Journal of Applied Mechanics*, 22:509–514, 1955.
- [76] F. Harlow and J. Welch. Numerical calculation of time-dependent viscous incompressible flow of fluid with free surface. *Physics of Fluids*, 8:2182–2189, 1965.
- [77] A. Harten, B. Engquist, S. Osher, and S. Chakravarthy. Uniformly high order accurate essentially non-oscillatory schemes, iii. *Journal of Computational Physics*, 71:231–303, 1987.
- [78] A. Harten and Osher. Uniformly high order accurate non-oscillatory schemes, i. *SIAM Journal on Numerical Analysis*, 24:279–309, 1987.
- [79] A. Harten, S. Osher, B. Engquist, and S. Chakravarthy. Some results on uniformly high-order accurate essentially non-oscillatory schemes. *Applied Numerical Mathematics*, 2:347–377, 1986.
- [80] D. Harvie and D. Fletcher. A new Volume of Fluid advection algorithm: The stream scheme. *Journal of Computational Physics*, 162:1–32, 2000.
- [81] D. Harvie and D. Fletcher. A new Volume of Fluid advection algorithm: The defined donating region scheme. *International Journal for Numerical Methods in Fluids*, 35:151–172, 2001.

- [82] A. Haselbacher and O. Vasilyev. Commutative discrete filtering on unstructured grids based on least-squares techniques. *Journal of Computational Physics*, 187:197–211, 2003.
- [83] J. Helmsen, P. Colella, and E. Puckett. Non-convex profile evolution in two dimensions using Volume of Fluid. Technical Report LBNL-40693, Lawrence Berkeley National Laboratory, 1997.
- [84] M. Herrmann. A Eulerian Level Set/vortex sheet method for two-phase interface dynamics. *Journal of Computational Physics*, 203:539–571, 2005.
- [85] M. Herrmann. Refined Level Set Grid method for tracking interfaces. Annual Research Briefs 2005 3–8, Center for Turbulence Research, 2005.
- [86] M. Herrmann. A balanced force Refined Level Set Grid method for two-phase flows on unstructured flow solver grids. Annual Research Briefs 2006 167–184, Center for Turbulence Research, 2006.
- [87] C. Hirt. An arbitrary Lagrangian-Eulerian method for incompressible flows. *Proceedings of the Second International Conference on Numerical Methods in Fluid Dynamics, Berkeley*, 1970.
- [88] C. Hirt, A. Amsden, and J. Cook. An arbitrary Lagrangian-Eulerian computing method for all speeds. *Journal of Computational Physics*, 14:227–253, 1974.
- [89] C. Hirt and B. Nichols. Volume of Fluid method for the dynamics of free boundaries. *Journal of Computational Physics*, 39:323–345, 1981.
- [90] J. Hong, T. Shinar, M. Kang, and R. Fedkiw. On boundary condition capturing for multiphase interfaces. *Journal of scientific Computing*, 31:99–125, 2007.

- [91] M. Hong. *Atomisation et mélange dans les jets coaxiaux liquide-gaz*. PhD thesis, Université de Grenoble, 2003.
- [92] J. Hoyt and J. Taylor. Turbulence structure in a water jet discharging in air. *Physics of Fluids*, 20:S253–S257, 1977.
- [93] J. Hoyt and J. Taylor. Waves on waterjets. *Journal of Fluid Mechanics*, 83:119–127, 1977.
- [94] C. Hu and C.-W. Shu. A discontinuous Galerkin finite element method for Hamilton-Jacobi equations. *SIAM Journal on Scientific Computing*, 21:666–690, 1999.
- [95] C. Hu and C.-W. Shu. Weighted essentially non-oscillatory schemes on triangular meshes. *Journal of Computational Physics*, 150:97–127, 1999.
- [96] K. Huh and D. Gosman. A phenomenological model of Diesel spray atomization. *Proceedings of International Conference on Multiphase Flow, Tsukuba, Japan*, 1991.
- [97] J. Hunt, A. Wray, and P. Moin. Eddies, streams and convergence zones in turbulent flows. Technical Report CTR-S88, Center for Turbulence Research, 1988.
- [98] E. Ibrahim. Spatial instability of a viscous liquid sheet. Technical Report 94-0562, American Institute of Aeronautics and Astronautics, 1994.
- [99] R. Issa. Solution of the implicitly discretised fluid flow equations by Operator-Splitting. *Journal of Computational Physics*, 62:40–65, 1985.
- [100] D. Jacqmin. An energy approach to the continuum surface tension method. Technical Report 96-0858, American Institute of Aeronautics and Astronautics, 1996.

- [101] D. Jacqmin. Calculation of two-phase Navier-Stokes flows using phase-field modeling. *Journal of Computational Physics*, 155:96, 1999.
- [102] D. Jacqmin. Contact-line dynamics of a diffuse fluid interface. *Journal of Fluid Mechanics*, 402:57–88, 2000.
- [103] H. Jasak. *Error analysis and estimation for the Finite Volume Method with application to fluid flows*. PhD thesis, Imperial College of Science, Technology & Medicine, 1996.
- [104] H. Jasak. Parallelisation and scalability in OpenFOAM. International Workshop on Scalable Engineering Software National Science Foundation, June 2010.
- [105] S. Jazayeri and X. Li. Non-linear instability of plane liquid sheets. *Journal of Fluid Mechanics*, 406:281–308, 2000.
- [106] G.-S. Jiang and D. Peng. Weighted ENO schemes for Hamilton–Jacobi equations. *SIAM Journal on Scientific Computing*, 21:2126–2143, 2000.
- [107] G.-S. Jiang and C.-W. Shu. Efficient implementation of weighted ENO schemes. *Journal of Computational Physics*, 126:202, 1996.
- [108] J. Kane. *Boundary-Element Analysis*. NJ: Prentice Hall, 1994.
- [109] M. Kang, R. Fedkiw, and X.-D. Liu. A boundary condition capturing method for multiphase incompressible flow. *Journal of Scientific Computing*, 15:323–360, 2000.
- [110] M. Käser and A. Iske. ADER schemes on adaptive triangular meshes for scalar conservation laws. *Journal of Computational Physics*, 205:486–508, 2005.
- [111] D. Kim, O. Desjardins, M. Herrmann, and P. Moin. The primary breakup of a round liquid jet by a coaxial flow of gas. *Proceedings of the 20th Annual Con-*

- ference of the Institute for Liquid Atomization and Spray Systems - Americas*, 2007.
- [112] R. Klein, S. Sadiki, and T. Janika. A digital filter based generation of in-flow data for spatially developing direct numerical simulation or large eddy simulations. *Journal of Computational Physics*, 186:652–665, 2003.
- [113] S. Kong, P. Senecal, and R. Reitz. Developments in spray modelling in diesel and direct-injection gasoline engines. *Oil & Gas Science and Technology - Revue de l'Institut Français du Pétrol*, 54:197–204, 1999.
- [114] R. Kraichnan. The structure of isotropic turbulence at very high Reynolds numbers. *Journal of Fluid Mechanics*, 5:497–543, 1959.
- [115] R. Kraichnan. Direct-interaction approximation for shear and thermally driven turbulence. *Physics of Fluids*, 7:1048–1062, 1964.
- [116] R. Kraichnan. Eulerian and Lagrangian renormalization in turbulence theory. *Journal of Fluid Mechanics*, 83:349–374, 1977.
- [117] A. Krommer and C. Ueberhuber. *Computational integration*. Society for Industrial and Applied Mathematics, 1998.
- [118] E. Labourasse, D. Lacanette, A. Toutant, P. Lubin, S. Vincent, O. Lebaigue, J.-P. Caltagirone, and P. Sagaut. Towards large eddy simulation of isothermal two-phase flows : Governing equations and a priori tests. *International Journal of Multiphase Flow*, 33:1–39, 2007.
- [119] B. Lafaurie, C. Nardonne, R. Scardovelli, S. Zaleski, and G. Zanetti. Modeling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, 113:134–147, 1994.
- [120] M. Lance and J. Bataille. Turbulence in the liquid phase of a uniform bubbly air-water flow. *Journal of Fluid Mechanics*, 222:95–118, 1991.

- [121] P. Lax and B. Wendroff. Systems of conservation laws. *Communications on Pure and Applied Mathematics*, 13:217–237, 1960.
- [122] A. Lefebvre. *Atomization and Sprays*. Hemisphere Publishing Corporation, 1989.
- [123] A. Leonard. Energy cascade in large-eddy simulations of turbulent fluid flows. *Advances in Geophysics*, 18A:237–248, 1974.
- [124] S. Leroux, C. Dumouchel, and M. Ledoux. The stability curve of Newtonian liquid jets. *Atomization and Sprays*, 6:623–647, 1996.
- [125] R. LeVeque. High-resolution conservative algorithm for advection in incompressible flow. *SIAM Journal on Numerical Analysis*, 33:627–665, 1996.
- [126] R. LeVeque and L. Lee. An immersed interface method for incompressible Navier-Stokes equations. *SIAM Journal on Scientific Computing*, 25:832–856, 2003.
- [127] R. LeVeque and Z. Li. The immersed interface method for elliptic equations with discontinuous coefficients and singular sources. *SIAM Journal on Numerical Analysis*, 31:1019–1044, 1994.
- [128] D. Lewis. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. *Proceedings of the Royal Society of London, Series A*, 202:81–96, 1950.
- [129] J. Li. Calcul d’interface affine par morceaux (piecewise linear interface calculation). *Comptes Rendus de l’Académie des Sciences Paris, Série IIb, (Paris)*, 320:391–396, 1995.
- [130] X. Li and R. Tankin. On the temporal instability of a two-dimensional viscous liquid sheet. *Journal of Fluid Mechanics*, 226:425–443, 1991.

- [131] X.-D. Liu, S. Osher, and T. Chan. Weighted Essentially Non-Oscillatory Schemes. *Journal of Computational Physics*, 115:200, 1994.
- [132] M. Longuet-Higgins and E. Cokelet. The deformation of steep surface waves on water, I A numerical method of computation. *Proceedings of the Royal Society of London, Series A*, 350:1–26, 1976.
- [133] J. Lopez, J. Hernandez, P. Gomez, and F. Faura. An improved PLIC-VOF method for tracking thin fluid structures in incompressible two-phase flows. *Journal of Computational Physics*, 208:51–74, 2005.
- [134] J. Lowengrub and I. Truskinovsky. Cahn-Hilliard fluids and topological transitions. *Proceedings of the Royal Society of London, Series A*, 454:2617, 1998.
- [135] A. Lozano and F. Barreras. Experimental study of the gas flow in an air-blasted liquid sheet. *Experiments in Fluids*, 31:367–376, 2001.
- [136] A. Lozano, F. Barreras, G. Hauke, and C. Dopazo. Longitudinal instabilities in an air-blasted liquid sheet. *Journal of Fluid Mechanics*, 437:143–173, 2001.
- [137] P. Macklin and J. Lowengrub. An improved geometry-aware curvature discretization for Level Set methods: Application to tumor growth. *Journal of Computational Physics*, 215:392–401, 2006.
- [138] J. Magnaudet, M. Rivero, and J. Fabre. Accelerated flows past a rigid sphere or a spherical bubble. part I Steady straining flow. *Journal of Fluid Mechanics*, 284:97–135, 1995.
- [139] S. Manservigi, S. Popinet, R. Scardovelli, and S. Zaleski. Two front-tracking algorithms for 2D interface problems. *Proceedings of the 3rd International Conference on Multiphase Flows*, 1998.
- [140] A. Mansour and N. Chigier. Disintegration of liquid sheets. *Physics of Fluids A*, 2:706–719, 1990.

- [141] A. Mansour and N. Chigier. Dynamic behaviour of liquid sheets. *Physics of Fluids A*, 3:2971–2980, 1991.
- [142] A. Mansour and N. Chigier. Effect of turbulence on the stability of liquid jets and resulting droplet size distributions. *Atomization and Sprays*, 4:583–604, 1994.
- [143] A. Mansour and T. Lundgren. Satellite formation in capillary jet breakup. *Physics of Fluids A*, 2:1141–1144, 1990.
- [144] L. Margolin, J. Reisner, and P. Smolarkiewicz. Application of the Volume-of-Fluid method to the advection-condensation problem. *Monthly Weather Review*, 125:2265–2273, 1997.
- [145] P. Marmottant. *Atomisation d'un liquide par un courant gazeux*. PhD thesis, Institut National Polytechnique de Grenoble, 2001.
- [146] P. Marmottant and E. Villermaux. On spray formation. *Journal of Fluid Mechanics*, 498:73–112, 2004.
- [147] A. Marsden, O. Vasilyev, and P. Moin. Construction of commutative filters for LES on unstructured meshes. *Journal of Computational Physics*, 175:584–603, 2002.
- [148] A. Massé. Experiments and numerical simulations of the flow within a model of a hydraulic turbine surge chamber. Master's thesis, McGill University, Montreal, 2010.
- [149] B. Mathieu, O. Lebaigue, and L. Tadriss. Dynamic contact line model applied to single bubble growth. *Proceedings of the 41st European Two-Phase Flow Group Meeting, May 2003*, 2003.
- [150] T. Menard. *Développement d'une méthode Level Set pour le suivi d'interface. Application à la rupture de jet liquide*. PhD thesis, Université de Rouen, 2007.

- [151] T. Menard, P.-A. Beau, S. Tanguy, and A. Berlemont. Primary break-up: DNS of liquid jet to improve atomization modelling. *Computational Methods in Multiphase Flow*, 3:343–352, 2005.
- [152] T. Menard, S. Tanguy, and A. Berlemont. Coupling Level Set/VOF/ghost fluid methods: Validation and application to 3D simulation of the primary break-up of a liquid jet. *International Journal of Multiphase Flow*, 33:510–524, 2007.
- [153] J. Meyer and D. Weihs. Capillary instability of an annular liquid jet. *Journal of Fluid Mechanics*, 179:531–545, 1987.
- [154] G. Miller and E. Puckett. A high-order Godunov method for multiple condensed phases. *Journal of Computational Physics*, 128:134–164, 1996.
- [155] J. Monaghan. Smoothed particle hydrodynamics. *Annual Review of Astronomy and Astrophysics*, 30:543–574, 1992.
- [156] J. Monaghan. Simulating free surface flows with SPH. *Journal of Computational Physics*, 110:399–406, 1994.
- [157] W. Mulder, S. Osher, and J. Sethian. Computing interface motion in compressible gas dynamics. *Journal of Computational Physics*, 100:209–228, 1992.
- [158] D. Nguyen, R. Fedkiw, and M. Kang. A boundary condition capturing method for incompressible flame discontinuities. *Journal of Computational Physics*, 172:71–98, 2001.
- [159] W. Noh and P. Woodward. SLIC (simple line interface calculation). *Proceedings of the Fifth International Conference on Numerical Methods in Fluid Dynamics Springer-Verlag New York*, 59:330–340, 1976.

- [160] M. Norman and K.-H. Winkler. *Astrophysical Radiation Hydrodynamics*, chapter 2-D Eulerian hydrodynamics with fluid interfaces, self-gravity and rotation, pages 187–221. New York: Reidel, 1986.
- [161] G. Nukiyama and Y. Tanasawa. Experiments on the atomization of liquids by means of air streams, parts II and IV. *Transactions of the Society of Mechanical Engineers Japan*, 5:63–75, 1939.
- [162] C. Ollivier-Gooch and M. Van Altena. A high-order-accurate unstructured mesh finite-volume scheme for the advection-diffusion equation. *Journal of Computational Physics*, 181:729–752, 2002.
- [163] E. Olsson and G. Kreiss. A conservative level set method for two phase flow. *Journal of Computational Physics*, 210:225–246, 2005.
- [164] E. Olsson, G. Kreiss, and S. Zahedi. A conservative level set method for two phase flow II. *Journal of Computational Physics*, 225:785–807, 2007.
- [165] OpenCFD Ltd. *OpenFOAM, Programmer’s Guide, version 1.5*, July 2008.
- [166] OpenCFD Ltd. *OpenFOAM, User Guide, version 1.5*, July 2008.
- [167] P. O’Rourke. *Collective drop effects on vaporizing liquid sprays*. PhD thesis, Princeton University, 1532-T, 1981.
- [168] P. O’Rourke and A. Amsden. The TAB method for numerical calculations of spray droplet breakup. Technical Paper 872089, Society of Automotive Engineers, 1987.
- [169] S. Osher and R. Fedkiw. *Level Set Methods and Dynamic Implicit Surfaces*. Applied Mathematical Sciences, 153. New York: Springer, 2002.
- [170] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

- [171] S. Osher and C.-W. Shu. High-Order Essentially Non-oscillatory schemes for Hamilton-Jacobi equations. *SIAM Journal on Numerical Analysis*, 28:907–922, 1991.
- [172] B. Parker and D. Youngs. Two and three dimensional Eulerian simulation of fluid flow with material interfaces. Technical Report 01/92, UK Atomic Weapons Establishment, Aldermaston, Berkshire, 1992.
- [173] D. Peng, B. Merriman, S. Osher, H. Zhao, and M. Kang. A PDE-based fast local Level Set method. *Journal of Computational Physics*, 155:410–438, 1999.
- [174] R. Penrose and A. Todd. On best approximate solutions of linear matrix equations. *Mathematical Proceedings of the Cambridge Philosophical Society*, 52:17–19, 1956.
- [175] C. Peskin. Numerical analysis of blood flow in the heart. *Journal of Computational Physics*, 25:220–252, 1977.
- [176] J. Pilliod. An analysis of piecewise linear interface reconstruction algorithms for Volume-of-Fluid methods. Master’s thesis, University of California, Davis, 1992.
- [177] J. Pilliod and E. Puckett. Second-order accurate Volume-of-Fluid algorithms for tracking material interfaces. Technical report, Lawrence Berkeley National Laboratory, 1997.
- [178] J. Pilliod and E. Puckett. Second-order accurate Volume-of-Fluid algorithms for tracking material interfaces. *Journal of Computational Physics*, 199:465–502, 2004.
- [179] J. Plateau. *Statique Expérimentale et Théorique des Liquides Soumis aux Seules Forces Moléculaires*. Gauthier-Villars, Paris, 1873.
- [180] S. Pope. *Turbulent Flows*. Cambridge University Press, 2000.

- [181] S. Popinet. http://gfs.sourceforge.net/wiki/index.php/main_page.
- [182] S. Popinet. A tree-based adaptive solver for the incompressible Euler equations in complex geometries. *Journal of Computational Physics*, 130:572–600, 2003.
- [183] S. Popinet. An accurate adaptive solver for surface-tension-driven interfacial flows. *Journal of Computational Physics*, 228:5838–5866, 2009.
- [184] S. Popinet and S. Zaleski. Simulation of axisymmetric free-surface viscous flow around a non-spherical bubble in the sonoluminescence regime. *Proceedings of the 3rd International Conference on Multiphase Flows*, 1998.
- [185] S. Popinet and S. Zaleski. A front tracking algorithm for the accurate representation of surface tension. *International Journal for Numerical Methods in Fluids*, 30:775–779, 1999.
- [186] T. Pringuey and R.S. Cant. High order schemes on 3D mixed-element unstructured meshes. Technical Report CUED/A-AERO/TR29, University of Cambridge, Department of Engineering, 2010.
- [187] T. Pringuey and R.S. Cant. High order schemes on three-dimensional general polyhedral meshes — Application to the level set method. *Communications in Computational Physics*, 12:1–41, 2012.
- [188] E. Puckett. A Volume-of-Fluid interface tracking algorithm with applications to computing shockwave refraction. *Proceedings of 4th International Symposium on Computational Fluid Dynamics*, pages 933–938, 1991.
- [189] E. Puckett, A. Almgren, J. Bell, D. Marcus, and W. Rider. A high order projection method for tracking fluid interfaces in variable density incompressible flows. *Journal of Computational Physics*, 130:269–282, 1997.
- [190] E. Puckett and J. Saltzman. A 3D adaptive mesh refinement algorithm for multimaterial gas dynamics. *Physica D*, 60:84–93, 1992.

- [191] R. Rangel and W. Sirignano. The linear and non-linear shear instability of a fluid sheet. *Physics of Fluids A*, 3:2392–2400, 1991.
- [192] J. Rayleigh. On the capillary phenomena of jets. *Proceedings of the Royal Society of London*, 29:71–97, 1879.
- [193] J. Rayleigh. On the stability of jets. *Proceedings of London Mathematical Society*, 10:4–13, 1879.
- [194] J. Rayleigh. Investigation of the character of the equilibrium of an incompressible heavy fluid of variable density. *Proceedings of the Royal Society of London*, 14:170–177, 1883.
- [195] L. Raynal. *Instabilité et entrainement à l'interface d'une couche de mélange liquide-gaz*. PhD thesis, Université Joseph Fourier, Grenoble, 1997.
- [196] R. Reitz. Modelling atomization processes in high-pressure vaporizing sprays. *Atomisation and Spray Technology*, 3:309–337, 1987.
- [197] R. Reitz and R. Diwakar. Structure of high-pressure fuel sprays. Technical Paper 870598, Society of Automotive Engineers, 1987.
- [198] Y. Renardy and M. Renardy. PROST: A parabolic reconstruction of surface tension for the Volume-of-Fluid method. *Journal of Computational Physics*, 183:400–421, 2002.
- [199] C. Rhie and W. Chow. Numerical study of the turbulent flow past an airfoil with trailing edge separation. *AIAA Journal*, 21:1525–1532, 1983.
- [200] E. Riber, V. Moureau, M. Garcia, T. Poinso, and O. Simonin. Evaluation of numerical strategies for large eddy simulation of particulate two-phase recirculating flows. *Journal of Computational Physics*, 228:539–564, 2009.

- [201] W. Rider and D. Kothe. Stretching and tearing interface tracking methods. *AIAA Computational Fluid Dynamics conference, 12th, San Diego CA, June 1995, Collection of Technical Papers*, (AIAA-1995-1717):806–816, 1995.
- [202] W. Rider and D. Kothe. Reconstructing volume tracking. *Journal of Computational Physics*, 141:112, 1998.
- [203] N. Rizk and A. Lefebvre. The influence of liquid film thickness on airblast atomization. *Transaction of ASME: Journal of Engineering for Power*, 102:706–710, 1980.
- [204] D. Rothman and S. Zaleski. Lattice-gas models of phase separation: interfaces, phase transitions, and multiphase flow. *Reviews of Modern Physics*, 66:1417–1479, 1994.
- [205] D. Rothman and S. Zaleski. *Lattice-Gas Cellular Automata*. Cambridge University Press, 1997.
- [206] G. Ruff, L. Bernal, and G. Faeth. Structure of the near-injector region of non-evaporating pressure-atomized sprays. *Journal of Propulsion and Power*, 7:221–230, 1991.
- [207] G. Ruff, P.-K. Wu, L. Bernal, and G. Faeth. Continuous- and dispersed-phase structure of dense non-evaporating pressure-atomized sprays. *Journal of Propulsion and Power*, 8:280–289, 1992.
- [208] H. Rusche. *Dispersed two-phase flows at high phase fractions*. PhD thesis, Imperial College of Science, Technology & Medicine, 2002.
- [209] G. Ryskin and L. Leal. Numerical solution of free boundary problems in fluid mechanics. part I The finite-difference technique. *Journal of Fluid Mechanics*, 148:1–17, 1984.

- [210] G. Ryskin and L. Leal. Numerical solution of free boundary problems in fluid mechanics. part II Buoyancy-driven motion of a gas bubble through a quiescent liquid. *Journal of Fluid Mechanics*, 148:19–35, 1984.
- [211] P. Sagaut. *Large Eddy Simulation for Incompressible Flows*. Springer-Verlag, 2003.
- [212] R. Saurel and R. Abgrall. A simple method for compressible multifluid flows. *SIAM Journal on Scientific Computing*, 21:1115–1145, 1999.
- [213] F. Savart. Mémoire sur la constitution des veines liquides lancées par des orifices circulaires en mince paroi. *Annales de Chimie et de Physique*, 53:337–398, 1833.
- [214] F. Savart. Mémoire sur le choc de deux veines liquides animées de mouvements directement opposés. *Annales de Chimie et de Physique*, 55:257–310, 1833.
- [215] F. Savart. Mémoire sur le choc d’une veine liquide lancée sur un plan circulaire. *Annales de Chimie et de Physique*, 54:56–87, 1833.
- [216] F. Savart. Suite du mémoire sur le choc d’une veine liquide lancée sur un plan circulaire. *Annales de Chimie et de Physique*, 54:113–145, 1833.
- [217] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 31:567–603, 1999.
- [218] R. Scardovelli and S. Zaleski. Interface reconstruction with least-square fit and split Eulerian-Lagrangian advection. *International Journal for Numerical Methods in Fluids*, 41:251–274, 2003.
- [219] R. Schulkes. The evolution and bifurcation of a pendant drop. *Journal of Fluid Mechanics*, 278:83–100, 1994.

- [220] U. Schumann. Subgrid scale model for finite difference simulations of turbulent flows in plane channels and annuli. *Journal of Computational Physics*, 18:376–404, 1975.
- [221] T. Schwartzkopff, M. Dumbser, and C. Munz. Fast high order ADER schemes for linear hyperbolic equations. *Journal of Computational Physics*, 197:532–539, 2004.
- [222] T. Schwartzkopff, C. Munz, and E. Toro. ADER: a high order approach for linear hyperbolic systems in 2D. *Journal of Scientific Computing*, 17:231–240, 2002.
- [223] P. Seppecher. Moving contact lines in the Cahn-Hilliard theory. *International Journal of Engineering Science*, 34:977–992, 1996.
- [224] J. Sethian. *An analysis of flame propagation*. PhD thesis, University of California, Berkeley, California; CPAM Rep. 79, 1982.
- [225] J. Sethian. Turbulent combustion in open and closed vessels. *Journal of Computational Physics*, 54:425–456, 1984.
- [226] J. Sethian. Curvature and the evolution of fronts. *Communications in Mathematical Physics*, 101:487–499, 1985.
- [227] J. Sethian. Numerical methods for propagating fronts. *Variational Methods for Free Surface Interfaces, Proceedings of the Sept, 1985 Vallambrosa Conference*, Eds. P. Concus and R. Finn, Springer-Verlag, NY, 1987.
- [228] J. Sethian. A fast marching Level Set method for monotonically advancing fronts. *Proceedings of the National Academy of Science USA*, 93:1591–1595, 1996.
- [229] J. Sethian. Fast marching methods. *SIAM Review*, 41:199–235, 1999.

- [230] C. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes. *Journal of Computational Physics*, 77:439–471, 1989.
- [231] C. Shu and S. Osher. Efficient implementation of essentially non-oscillatory shock-capturing schemes, II. *Journal of Computational Physics*, 83:32–78, 1989.
- [232] C. Siegler, A. Lozano, F. Barreras, and D. Low. The effects of sheet thickness on the oscillation of an air-blasted liquid sheet. *Experiments in Fluids*, 39:127–139, 2005.
- [233] H. Simmons. The correlation of drop-size distributions in fuel nozzle sprays. *Transactions of ASME: Journal of Engineering for Power*, 99:309–319, 1977.
- [234] J. Smagorinsky. General circulation experiments with the primitive equations: I. The basic equations. *Monthly Weather Review*, 91:99–164, 1963.
- [235] H. Squire. Investigation of the instability of a moving film. *British Journal of Applied Physics*, 4:167–169, 1953.
- [236] B. Stapper and G. Samuelsen. An experimental study of the breakup of a two-dimensional liquid sheet in the presence of co-flow air shear. Technical Report 89-0461, American Institute of Aeronautics and Astronautics, 1990.
- [237] B. Stapper, W. Sowa, and G. Samuelsen. An experimental study of the effects of liquid properties on the breakup of a two-dimensional liquid sheet. *ASME International Gas Turbine and Aeroengine Congress and Exposition, Belgium*, 1990.
- [238] G. Strang. On the construction and comparison of difference schemes. *SIAM Journal on Numerical Analysis*, 5:506, 1968.

- [239] A. Stroud. *Approximate calculation of multiple integrals*. NJ: Prentice-Hall, 1971.
- [240] M. Sussman. A second order coupled Level Set and Volume-of-Fluid method for computing growth and collapse of vapor bubbles. *Journal of Computational Physics*, 187:110–136, 2003.
- [241] M. Sussman, A. Almgren, J. Bell, P. Colella, L. Howell, and M. Welcome. Adaptive Level Set approach for incompressible two-phase flows. *Journal of Computational Physics*, 148:81–124, 1999.
- [242] M. Sussman, E. Fatemi, P. Smereka, and S. Osher. An improved Level Set method for incompressible two phase flows. *Computers and Fluids*, 27:663–680, 1998.
- [243] M. Sussman and E. Puckett. A coupled Level Set and Volume-of-Fluid method for computing 3D and axisymmetric incompressible two-phase flows. *Journal of Computational Physics*, 162:301–337, 2000.
- [244] M. Sussman, P. Smereka, and S. Osher. A Level Set approach for computing solutions to incompressible two phase flow. *Journal of Computational Physics*, 114:146–159, 1994.
- [245] M. Sussman, K. Smith, M. Hussaini, M. Ohta, and R. Zhi-Wei. A sharp interface method for incompressible two-phase flows. *Journal of Computational Physics*, 221:469–505, 2007.
- [246] S. Tanguy. *Développement d’une méthode de suivi d’interface. Applications aux écoulements diphasiques*. PhD thesis, Université de Rouen, 2004.
- [247] S. Tanguy, T. Ménard, and A. Berlemont. A Level Set Method for vaporizing two-phase flows. *Journal of Computational Physics*, 221:837–853, 2007.

- [248] F. Tanner. Liquid jet atomization and droplet breakup modelling of non-evaporating diesel fuel sprays. *Society of Automotive Engineers Transactions, Journal of Engines*, 106:127–140, 1997.
- [249] G. Taylor. The instability of liquid surfaces when accelerated in a direction perpendicular to their planes. *Proceedings of the Royal Society of London, Series A*, 201:192–196, 1950.
- [250] G. Taylor. The dynamics of thin sheets of fluid I. Water bells. *Proceedings of the Royal Society of London, Series A*, 253:289–295, 1959.
- [251] V. Titarev and D. Drikakis. Uniformly high order schemes on arbitrary unstructured meshes for advection-diffusion equations. *Computers & Fluids*, 46:467–471, 2011.
- [252] V. Titarev and E. Toro. ADER: arbitrary high order Godunov approach. *Journal of Scientific Computing*, 17:609–618, 2002.
- [253] V. Titarev and E. Toro. ADER schemes for three-dimensional non-linear hyperbolic systems. *Journal of Computational Physics*, 204:715–736, 2005.
- [254] V. Titarev, P. Tsoutsanis, and D. Drikakis. WENO schemes for mixed-element unstructured meshes. *Communications in Computational Physics*, 8:585–609, 2010.
- [255] E. Toro. *Riemann solvers and numerical methods for fluid dynamics: a practical introduction*. Springer, Verlag, 1999.
- [256] E. Toro, R. Millington, and L. Nejad. *Godunov methods. Theory and Applications*, chapter Towards very high order Godunov schemes, pages 905–938. Kluwer Academic Publishers, 2001.

- [257] E. Toro and V. Titarev. Solution of the generalized Riemann problem for advection-reaction equations. *Proceedings of the Royal Society of London, Series A*, 458:271–281, 2002.
- [258] E. Toro and V. Titarev. ADER schemes for scalar hyperbolic conservation laws with source terms in three space dimensions. *Journal of Computational Physics*, 202:196–215, 2005.
- [259] G. Tryggvason, B. Bunner, A. Esmaeeli, D. Juriv, N. Al-Rawahi, W. Tauber, J. Han, J. Nas, and Y.-J. Jan. A front-tracking method for the computations of multiphase flow. *Journal of Computational Physics*, 169:708–759, 2001.
- [260] G. Tryggvason and S. Unverdi. Computations of three-dimensional Rayleigh-Taylor instability. *Physics of Fluids A*, 2:656, 1990.
- [261] S. Unverdi and G. Tryggvason. A front tracking method for viscous incompressible multi-fluid flows. *Journal of Computational Physics*, 100:25–37, 1992.
- [262] O. Ushakova. Conditions of nondegeneracy of three-dimensional cells. A formula of a volume of cells. *SIAM Journal on Scientific Computing*, 23:1274–1290, 2001.
- [263] S. Van der Pijl, A. Segal, A. Vuik, and P. Wesseling. A mass-conserving Level Set method for modelling multi-phase flows. *International Journal for Numerical Methods in Fluids*, 7:339–361, 2005.
- [264] J. Van der Waals. Thermodynamische theorie der capillariteit in de onderstelling van continue dichtheidsverandering. *Verhandelingen der Koninklijke Akademie van Wetenschappen te Amsterdam*, 1(8), 1893.
- [265] O. Vasilyev, T. Lund, and P. Moin. A general class of commutative filters for LES in complex geometries. *Journal of Computational Physics*, 146:82–104, 1998.

- [266] E. Villermaux and C. Clanet. Life of a flapping liquid sheet. *Journal of Fluid Mechanics*, 462:341–363, 2002.
- [267] B. Vreman. *Direct and large eddy simulation of the compressible turbulent mixing layer*. PhD thesis, Twente University, 1995.
- [268] B. Vreman, B. Geurts, and J. Kuerten. A priori tests of large eddy simulation of the compressible plane mixing layer. *Journal of Engineering Mathematics*, 29:299–327, 1995.
- [269] R. Wang and R. Spiteri. Linear instability of the fifth-order WENO method. *SIAM Journal on Numerical Analysis*, 45:1871–1901, 2007.
- [270] C. Weber. Zum zerfall eines flüssigkeitsstrahles. *Zeitschrift fuer Angewandte mathematik und Mechanik*, 11:136–141, 1931.
- [271] H. Weller. <http://www.opencfd.co.uk/openfoam/index.html#openfoam>.
- [272] H. Weller. A code independent notation for finite volume algorithms. Technical Report TR/HGW/01, Nabla Ltd, 2002.
- [273] M. Williams, D. Kothe, and E. Puckett. *Fluid Dynamics Interfaces*, chapter Convergence and accuracy of continuum surface tension models, pages 294–305. Cambridge University Press, 1999.
- [274] P.-K. Wu and G. Faeth. Aerodynamic effects on primary breakup of turbulent liquids. *Atomization and Sprays*, 3:265–289, 1993.
- [275] P.-K. Wu and G. Faeth. Onset and end of drop formation along the surface of turbulent liquid jets in still gases. *Physics of Fluids*, 7:2915–2917, 1995.
- [276] P.-K. Wu, G. Ruff, and G. Faeth. Primary breakup in liquid-gas mixing layers. *Atomization and Sprays*, 1:421–440, 1991.

- [277] P.-K. Wu, L.-K. Tseng, and G. Faeth. Primary breakup in gas/liquid mixing layers for turbulent liquids. *Atomization and Sprays*, 2:295–317, 1992.
- [278] Y. Yi and R. Reitz. Modeling the primary break-up of high-speed jets. *Atomization and Sprays*, 14:53–80, 2004.
- [279] J. York, H. Stubbs, and M. Tek. The mechanisms of disintegration of liquid sheets. *Transactions of ASME*, pages 1279–1286, 1953.
- [280] A. Yoshizawa. A statistically-derived subgrid model for the large-eddy simulation of turbulence. *Physics of Fluids*, 25:1532–1538, 1982.
- [281] A. Yoshizawa. A statistically-derived subgrid-scale kinetic energy model for the large-eddy simulation of turbulent flows. *Journal of the Physical Society of Japan*, 54:2834–2839, 1985.
- [282] A. Yoshizawa. Statistical theory for compressible turbulent shear flows, with the application to subgrid modeling. *Physics of Fluids*, 29:2152–2164, 1986.
- [283] T. Young. An essay on the cohesion of fluids. *Philosophical Transactions of the Royal Society of London*, 95:65–87, 1805.
- [284] S. Zalesak. Fully multidimensional flux-corrected transport algorithms for fluids. *Journal of Computational Physics*, 31:335–362, 1979.
- [285] Y.-T. Zhang and C.-W. Shu. High order WENO schemes for Hamilton–Jacobi equations on triangular meshes. *SIAM Journal on Scientific Computing*, 24:1005–1030, 2003.
- [286] Y.-T. Zhang and C.-W. Shu. Third order WENO scheme on three dimensional tetrahedral meshes. *Communications in Computational Physics*, 5:836–848, 2009.