



Università
degli Studi
della Campania
Luigi Vanvitelli

Engineering
Department

Master of Science in
Mechanical Engineering

Thesis in Fluid Dynamics
**Numerical investigation of the
aerobreakup of water columns using
OpenFOAM**

Candidate
Felice Edoardo Taglialatela
A19/467

Supervisor:
Prof. Giuliano De Stefano

A. Y. 2023/2024

Numerical investigation of the aerobreakup of water columns using OpenFOAM, © October 2024

Author:

Felice Edoardo TAGLIALATELA

Supervisor:

Prof. Giuliano DE STEFANO

Institute:

Università degli Studi della Campania Luigi Vanvitelli

CONTENTS

List of Figures	v
List of Tables	vii
Abstract	viii
Acknowledgments	ix
List of abbreviations	x
1 INTRODUCTION	1
1.1 Thesis Outline	1
2 PHYSICAL PROBLEM	2
2.1 Rain erosion	2
2.2 Approaches to fluid dynamical problems	3
2.3 Problem description	4
2.4 Compressible aerodynamics	5
2.5 Navier-Stokes equations	6
2.6 Governing multiphase equations	7
2.7 Closure	8
3 NUMERICAL INVESTIGATION	10
3.1 Mathematical classification of flows	10
3.1.1 Hyperbolic flows	10
3.2 Computational Fluid Dynamics	11
3.3 Possibilities and limitations of numerical methods	11
3.4 Hybrid method	12
3.4.1 VOF method	12
3.4.2 ACID approach	13
3.4.3 Finite Volume Discretization	14
3.4.4 Hybrid approximation of fluxes	15
3.4.5 Pressure equation	16
3.4.6 Liquid phase volume fraction transport equation	17
3.4.7 The overall numerical algorithm	18
3.4.8 Workflow	19
3.4.9 Note: MULES limiter	20
4 APPLIED CFD	21

4.1	About OpenFOAM	21
4.1.1	OpenFOAM case structure	21
4.2	Pre-processing	23
4.2.1	Mesh generation workflow	23
4.2.2	Boundary conditions	25
4.2.3	Non-Reflective Boundary Conditions	25
4.3	Solver	27
4.3.1	hybridCentralSolvers	27
4.4	Numerical scheme settings	27
4.5	Post-processing	27
4.5.1	Coded drift function	28
5	RESULTS	30
5.1	Validation	30
5.1.1	Moving contact - 1D Riemann Problem	30
5.1.2	Shock tube	31
5.1.3	Experimental comparison	34
5.2	Numerical comparison	45
5.3	Results	48
5.3.1	Kelvin-Helmholtz instabilities and Shear-Induced Entrainment	53
5.4	Note: thermodynamic conditions of water	55
5.5	Conclusions	56
	BIBLIOGRAPHY	57

LIST OF FIGURES

Figure 2.1	Rain erosion on wind turbine blades (left) and effects on two different Titanium-Alluminium alloys (right) [6].	3
Figure 3.1	Schematic illustration of the colour function $\psi = \alpha$ around cell (i, j) without (left) and with Acoustically Conservative Interface Discretization (ACID) technique (right) [21].	14
Figure 3.2	Different arrangements of finite volume grids.	15
Figure 3.3	Workflow of a typical Computational Fluid Dynamics (CFD) simulation loop.	19
Figure 4.1	A single block.	23
Figure 4.2	Unrefined mesh, 1 <i>CellsPerDiameter</i> (CPD).	23
Figure 4.3	Mesh after refinement.	24
Figure 4.4	Comparison between standard BCs (1) and NRBC (2) [27].	26
Figure 5.1	Moving contact discontinuity: volume fraction distribution.	30
Figure 5.2	Shock tube setup. <i>Driver</i> (4) and <i>driven</i> section (1).	31
Figure 5.3	Diaphragm ratio and pressure shock ratio for $M = 1.47$	31
Figure 5.4	Mesh of the shock tube problem, resolution: 16 CPD.	32
Figure 5.5	Contour of pressure at $t = 0\text{s}$, driver (red) and driven region (blue).	32
Figure 5.6	Mesh independence study and comparison with theoretical results.	33
Figure 5.7	Experimental setup.	34
Figure 5.8	Details of the simulation setup at $t = 0\text{s}$; iso-alphas are shown in black for $0.0001 < \alpha < 0.99$	35
Figure 5.9	Mesh I	35
Figure 5.10	Mesh II	36
Figure 5.11	Mesh III	36
Figure 5.12	Leading edge drift normalized by the initial diameter.	37
Figure 5.13	Spanwise deformation normalized by the initial diameter.	37
Figure 5.14	Impact of the shock wave on the surface of the cylinder, $0.001 < \alpha < 0.9$	38
Figure 5.15	Double exposure holographic interferogram from the experiment on the left side ($t_1 = 23\mu\text{s}$, $t_2 = 43\mu\text{s}$), numerical Schlieren on the right.	39
Figure 5.16	Gas phase isopycnics by Igara and Takayama (left, $t_1 = 23\mu\text{s}$, $t_2 = 43\mu\text{s}$) and computed (right).	40
Figure 5.17	Pressure and Schlieren (left); velocity and vorticity magnitude (right), from top to bottom, at $t_1^* = 0.235$, $t_2^* = 1.176$, $t_3^* = 3.057$	42

Figure 5.18	Contours of pressure and Schlieren (left) and velocity/vorticity magnitude (right), from top to bottom, at $t_1^* = 5.88, t_2^* = 10.584, t_3^* = 12.936$	43
Figure 5.19	Contours of pressure and Schlieren (left) and velocity/vorticity magnitude (right), from top to bottom, at $t_1^* = 16.229, t_2^* = 19.522, t_3^* = 21.404$	44
Figure 5.20	Comparison with Meng and Colonius numerical results.	45
Figure 5.21	Positive and negative vorticity streams interacting to form a recirculation region; reference on top and computed results on the bottom.	45
Figure 5.22	Velocity vectors and vortices on the <i>edges</i> of the cylinder.	46
Figure 5.23	From top to bottom: Kelvin-Helmholtz (KH) instabilities (1), ligaments and lips formation (2) and ligaments breakup (3); contour of liquid volume fraction α from 1 (red) to 0 (blue).	47
Figure 5.24	Normalized drift of the front column.	48
Figure 5.25	Normalized deformation of the front column.	48
Figure 5.26	Normalized drift of the rear column.	48
Figure 5.27	Normalized deformation of the rear column.	48
Figure 5.28	Unsteady C_D	50
Figure 5.29	Fitting curves.	50
Figure 5.30	$t_1^* = 3.137, t_2^* = 6.335, t_3^* = 12.578$	51
Figure 5.31	$t_1^* = 15.919, t_2^* = 25.544, t_3^* = 30.885$	52
Figure 5.32	Vorticity on the $z = 0$ plane, $t^* = 2.10$	53
Figure 5.33	Vorticity on the $z = 0$ plane, $t^* = 6.9$	54
Figure 5.34	Vorticity on the $z = 0$ plane, $t^* = 11.05$	54

LIST OF TABLES

Table 2.1	Phase properties.	8
Table 4.1	Boundary conditions of the Initial Boundary Value Problem (IBVP). .	25
Table 4.2	Non-Reflective Boundary Conditions (NRBC)s proposed.	27
Table 4.3	Used softwares.	29
Table 5.1	Boundary conditions.	30
Table 5.2	Boundary conditions for the Shock tube problem.	32

ABSTRACT

This thesis work aims to develop a virtual wind tunnel using an alternative solver developed for OpenFoam to accurately predict the physical phenomena associated with the breakup of rain droplets induced by a normal shock wave. The primary focus is on understanding the degree of droplet distortion caused by high-speed flows encountered in transonic and hypersonic flight. Using a physics-based methodology, the project aims to predict significant changes in the shape of the water columns, which is crucial to assess the structural damage caused by rain erosion. Accurate predictions can lead to more efficient experimental assessments, reduced costs, and improved structural designs with less oversizing, overall lighter systems, and lower fuel consumption.

Rain erosion poses significant challenges to the stability and efficiency of commercial aircraft, wind turbines, and supersonic UAVs, with damage increasing exponentially with velocity. Previous studies have primarily been experimental, involving shock tubes and wind tunnels, to understand droplet drift, deformation, and fragmentation under high-speed flows. Numerical studies have also been conducted, with various solvers attempting to replicate the shock wave-droplet interaction. However, these studies often face limitations due to high computational costs, leading to approximations in the computational domain. This project will address these limitations by utilizing the capabilities of OpenFoam, an open-source software, to replicate the phenomenon and provide a comprehensive and reliable solution for predicting the behavior of water columns undergoing a shock wave load.

Keywords: OpenFOAM; multiphase flow; computational fluid dynamics (CFD); vofTwo-PhaseCentralFoam; KNP method; aerobreakup; 2D

ACKNOWLEDGMENTS

I would like to express my profound gratitude to Матвей Крапотин, the developer of *vofTwoPhaseCentralFoam*, the solver used in this thesis. His insightful discussions and guidance greatly enriched the entire research experience. Furthermore, I am deeply thankful to my supervisor for allowing me the freedom to explore my interests and for providing invaluable direction throughout this journey. This support will resonate with me for years to come.

LIST OF ABBREVIATIONS

The following abbreviations are used in this manuscript:

- VOF Volume of Fluid
CFD Computational Fluid Dynamics
EOS Equation Of State
RANS Reynolds Averaged Navier-Stokes
LES Large Eddy Simulations
GPL General Public License
GUI Graphical User Interface
NRBC Non-Reflective Boundary Conditions
PDE Partial Differential Equation
PDEs Partial Differential Equations
IBVP Initial Boundary Value Problem
GRP Generalized Riemann Problem
KNP Kurganov-Noelle-Petrova
HLL Harten-Lax-van Leer
HLLC Harten-Lax-van Leer-Contact
ACID Acoustically Conservative Interface Discretization
PIMPLE Pressure Implicit with Splitting Operators and Semi-Implicit Pressure-Linked Equation
FVM Finite Volume Method
GFM Ghost-Fluid Method
PISO Pressure-Implicit with Splitting of Operators
SIMPLE Semi-Implicit Method for Pressure Linked Equations
CFL Courant-Friedrichs-Levy
TVD Total Variation Diminishing
CPD Cells Per Diameter
LOT Lower Order Terms
MULES Multidimensional Universal Limited Explicit Solver
SIE Shear-Induced Entrainment
RTP Rayleigh-Taylor Piercing
KH Kelvin-Helmholtz

INTRODUCTION

1.1 THESIS OUTLINE

The remainder of this thesis is organized as follows.

Chapter 2 offers an overview of the effects related to the high-speed collision of rain droplets with supersonic vehicles, and discusses its implications for existing industrial applications.

Chapter 3 contains the details of the numerical model used to capture the mechanism of deformation of the water column, subject to high-speed airflow [1]. It also provides background informations on applied CFD [2], being the technique used in this work.

Chapter 4 describes OpenFOAM, the open source software used to simulate the interaction, and outlines the process followed to achieve the results, focusing on the pre-processing and post-processing part.

Chapter 5 is where the numerical framework is validated and the results obtained are presented and compared to previous studies; moreover, the interaction mechanism and the subsequent breakup are thoroughly described with the support of the said results.

2

PHYSICAL PROBLEM

2.1 RAIN EROSION

When liquid drops are suddenly exposed to high speed flow, they deform and shatter. This phenomenon is called **aerobreakup**. The study of droplet breakup has several applications, ranging from combustion and detonation of multiphase mixtures to rain erosion damage [3] to reentry vehicles, supersonic aircrafts and missiles. For propeller aircraft, rain was mainly a vision problem, but in the jet age planes were vulnerable to raindrops eroding paint coatings and damaging plastic, ceramic and even metal components. The problem was even more acute with rockets; for a spacecraft re-entering the atmosphere at high velocity, every droplet is a liquid bullet. For high speeds, researchers use compressed gas guns to fire test articles into water droplets suspended in webs. They developed extreme water pistols [4] firing droplets at supersonic speeds, simulating flight through rain. The primary concern of this study is to numerically investigate the phenomena associated with droplet breakup.

High speed flight through natural weather is a significant challenge, let alone consider adverse weather. The severity of the environment increases exponentially with velocity. The problem arises when such bodies produce a shock wave that impacts the rain. The effect of this interaction could damage the nosetip of the aforementioned structures, which are commonly manufactured using brittler materials, compared to the rest of the body. The key concept is that the impact force and therefore the vehicle damage is a function of the droplet state at the moment of impact. The uniqueness of water damage lies in the significant changes in shape and mass that make it very difficult to assess the potential damage [5].

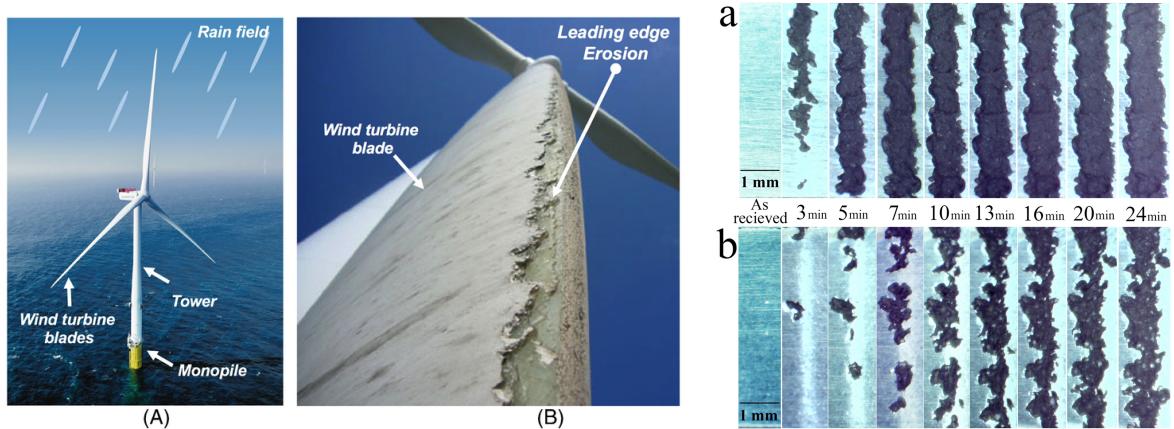


Figure 2.1: Rain erosion on wind turbine blades (**left**) and effects on two different Titanium-Aluminium alloys (**right**) [6].

2.2 APPROACHES TO FLUID DYNAMICAL PROBLEMS

Fluid dynamical problems are far from easy to solve. One has to make many assumptions to simplify the equations and obtain an analytical solution; however, they generally have no relevance in engineering problems.

The most common approach involves a combination of approximations and dimensional analysis. For example, dimensional analysis shows that the drag force on a certain body depends on the frontal area of the said object and the dynamic pressure:

$$D = C_D S \rho u_\infty^2 \quad (2.1)$$

Another example is the (previously mentioned) infamous Reynolds number

$$Re = \frac{\rho u L}{\mu} \quad (2.2)$$

that helps predict fluid flow patterns in different situations by measuring the ratio between inertial and viscous forces. At low Reynolds numbers, flows tend to be dominated by laminar (sheet-like) flow, while at high Reynolds numbers, flows tend to be turbulent.

This approach is very useful when the system is well described by one or two parameters, so complex geometries are not applicable. The problem is that many flows require several dimensionless parameters to be correctly and sufficiently described. However, it is nearly impossible to set up an experiment that correctly scales all the quantities. Examples are fluids around aircrafts: if the model is scaled down, velocity needs to be increased, and it may result in a too high Mach number if the same fluid is used; one has to find a fluid to match both parameters (Reynolds and Mach). It is the same for ships, but with the Froude¹ and Reynolds numbers. The other problem with experiments is to accurately measure the parameters of interest. Some quantities, though, are simply not measurable nowadays,

¹ $Fr = \frac{u}{\sqrt{gL_0}}$ is a dimensionless number defined as the ratio of the flow inertia to the external force field

or measurable with insufficient accuracy. In many cases, details are important and it might be necessary to know where the flow separates or where the temperature exceeds some fixed value.

An alternative came with the birth of computers. Although numerical methods used to solve partial differential equations already existed at the time, they were of little use before computers arrived. The performance of these machines increased exponentially (Moore's law²) and only favors the end users. While the first computers performed only a few hundred operations per second (and they made us reach the Moon!), machines are now being designed to produce **teraflops** = 10^{12} floating point operations per second. Storage capacity has also dramatically increased, in both quantity and speed.

Once the power of computers had been recognized, interest in numerical methods greatly arose. Solution of fluid mechanics equations on computers has become so important and frequent that it now occupies the attention of roughly a third of all researchers in fluid mechanics. And the number is still increasing. This field is known as CFD.

2.3 PROBLEM DESCRIPTION

The study of aerobreakup is very complex and really benefits from dimensional analysis. The droplet breakup is known to intensify with increasing Weber number [7]. Calling ρ_g the freestream density, u the freestream velocity, d some reference length and σ the surface tension:

$$We = \frac{Drag}{Cohesion} = \frac{\rho_g u^2 d}{\sigma} \quad (2.3)$$

The different breakup regimes are characterized by a given range of the said nondimensional number, and the transitions are marked by critical values of it. The different regimes embody the governing physics and define the quantitative features involved. Breakup has classically been divided by *Reinecke* [8] in five distinct regimes, before being re-classified by *Theofanous* and *Li* in [7] in the following primary ones:

- Rayleigh-Taylor Piercing (RTP)
- Shear-Induced Entrainment (SIE), $We > 10^3$

Here, the density and the velocity are those of the shocked gas, and the length is the initial droplet diameter. Using the experimental values for a water droplet in air, a surface tension of $\sigma = 0.07 N/m$, and a dynamic viscosity of $\mu = 1.8 \times 10^{-5} Pa \cdot s$ the approximate Weber and Reynolds numbers corresponding to $M = 1.47$ are

$$We = 7.3 \times 10^3 \quad \& \quad Re = 1.314 \times 10^5 \quad (2.4)$$

These high Weber and Reynolds numbers allow us to neglect surface tension and viscosity, for the early stages of the breakup.

The re-classification of the breakup modes involved the so-called *catastrophic regime*. It was

² The *law* is the observation that the number of transistors in an integrated circuit doubles about every two years. *Moore's law* is an observation and projection of a historical trend.

argued to be an artifact of the poor quality of the shadowgraph visualization³. Certain studies have also tried to correlate breakup with parameters such as the density or viscosity ratios of fluids. Various studies have attempted to calculate the acceleration of the droplet [9–11], it is often used a polynomial fitting to approximate the velocity and then differentiate to obtain the acceleration.

The implication of droplet breakup is that once the droplet has stripped to zero mass, it no longer presents a threat to the structure. Much of the previous literature consists of experimental work performed in shock tubes; numerical studies have been conducted on in-house solvers or commercial CFD frameworks. Moreover, they often make use of existing symmetries to lighten the computational demand. Although the normal shock per se has little to no effect on the droplet, generating it by setting up a shock tube-like configuration is a reliable way to obtain the high-speed conditions around the droplet. To do it, *Gary A. Sod* studies are used⁴. The results are presented in the dedicated chapter.

2.4 COMPRESSIBLE AERODYNAMICS

In the daily life we commonly observe *incompressible fluid dynamics*: problems regarding water, so-called *stiff* fluids or low speed flows, in comparison with the speed of sound in that fluid. On the other hand, when studying high-speed gas dynamics, the compressibility effect becomes nonnegligible and we have to make a distinction; therefore, we study what we call *compressible fluid dynamics*. For instance, in modern aircraft design (from the 1969 Concorde to the most modern Lockheed Martin X-59), the speed can be several times higher than the speed of sound. Such compressibility effects may dominate the phenomenon and make the system way more complex.

The flow regime is characterized by a non dimensional number called *Mach number*, M , which is the ratio of the speed of the aircraft to the local speed of sound:

$$M = \frac{v}{a} \quad (2.5)$$

Speed of sound is chosen as reference since it is the speed at which small disturbances are transmitted through the gas, as a result of collisions between the randomly moving molecules. The transmission of a small disturbance through a gas is an isentropic process, since the conditions in the gas are the same before and after the disturbance passes through. The speed of sound in a certain gas is constant and can be calculated starting from mass and momentum conservation, which shows it to be

$$a = \sqrt{\gamma RT} \quad (2.6)$$

Aeronautical engineers use Mach number to distinguish between various flow regimes:

- **subsonic** if $M < 1$, and if $M < 0.3$ compressibility can be neglected

³ Shadowgraph is an optical method that reveals non-uniformities in transparent media; it is similar to the Schlieren method, that will be presented later in the manuscript

⁴ *Sod shock tube* is also a benchmark to test numerical models when dealing with compressible problems.

- **transonic** when $M \sim 1$: at some places on the object, the local speed exceeds the speed of sound. Compressibility effects are most important in transonic flows as the drag increases (that is why aircraft do not fly around $M = 1$)
- **supersonic** for $1 < M < 3$: compressibility effects are important and shock waves are generated; for $3 < M < 5$, aerodynamic heating also becomes very important
- **hypersonic** if $M > 5$; the chemistry of the gas must be taken into account when calculating forces since chemical reaction are enforced. For space shuttles, coming towards the Earth at $M \sim 25$, air becomes ionized plasma and a special insulation must be realized

A crucial characteristic of supersonic flow is that perturbations **do not propagate in the upstream direction**.

2.5 NAVIER-STOKES EQUATIONS

The equations describing the motion of viscous fluids are the infamous Navier-Stokes equations, a set of coupled, nonlinear Partial Differential Equations (PDEs), deriving from the conservation of mass and from momentum balance. In this paragraph a quick overview is given, where each of the equations is obtained using the divergence theorem, to write them in the so-called *local form*. The first equation is the *continuity*, which represents the local conservation of the mass. Calling ρ the density of the fluids, the equation can be written as follows:

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.7)$$

The differential form of the continuity equation is coordinate free. This form can be transposed into a specific coordinate system by providing the expression for the divergence operator in that coordinate system.

The second vector equation is obtained from Newton's second law applied to a fluid, where the sum of the forces has to be expressed in terms of intensive properties. The surface forces due to pressure p and stresses T represent, from the microscopic point of view, the momentum fluxes across a surface. To express those microscopic fluxes in terms of density and velocity, some assumptions have to be made. The simplest one is that the fluid is *Newtonian*, therefore, the stress tensor T , which is the molecular rate of transport of momentum, can be written as stated in the previous chapter. With μ being the dynamic viscosity, I the unit tensor and E the strain rate tensor, the stress tensor in cartesian coordinates is

$$T_{ij} = -\left(p + \frac{2}{3}\mu \frac{\partial u_j}{\partial x_i}\right)\delta_{ij} + 2\mu E_{ij} \quad (2.8)$$

$$E_{ij} = \frac{1}{2} \left(\frac{\partial u_j}{\partial x_i} + \frac{\partial u_i}{\partial x_j} \right) \quad (2.9)$$

For incompressible flows, equation 2.8 simplifies since the second term in the brackets is zero. The corresponding momentum equation, including the body forces, is:

$$\rho \left(\frac{\partial \mathbf{u}}{\partial t} + \mathbf{u} \cdot \nabla \mathbf{u} \right) = -\nabla p + \rho \mathbf{g} + \nabla \cdot \mathbf{S} \quad (2.10)$$

The momentum equations are said to be in **strong conservation form** if all terms have the form of the divergence of a vector or tensor. This is possible when fixed directions are used, otherwise, apparent forces have to be included (and they are non conservative). The strong conservative form, when used with a Finite Volume Method (FVM), automatically insures global momentum conservation in the calculation. Equation 2.10 is in a non-conservative form due to the presence of the pressure term.

The third and last is the energy balance of the fluid, accounting for the change in internal energy due to convection, conduction and viscous effects:

$$\frac{\partial}{\partial t} \left(\rho e + \frac{1}{2} \rho u^2 \right) + \nabla \cdot \left(\left(\rho e + p + \frac{1}{2} \rho u^2 \right) \mathbf{u} \right) = \nabla \cdot (k \nabla T) + \Phi \quad (2.11)$$

where Φ is the viscous dissipation term. These are single-phase equations, and the addition of another phase implies a non-trivial degree of complexity. The more sophisticated model will be presented in the following section.

2.6 GOVERNING MULTIPHASE EQUATIONS

The multiphase flow is governed by the Navier-Stokes equations 2.5, adapted for multiple components. Each phase is considered immiscible, and, when mass transfer and surface tension are not present, the surface advection is governed by the local flow velocity. The following **reduced model** has been proposed by *Kapila et al.* [12]. It shares a fair degree of similarity with the **five equation model** by *Allaire et al.* [13], simplifying the terms where one phase is largely incompressible, making it suitable for simulations involving air-water interactions. This model focuses on reducing the computational complexity while maintaining essential physical features of the flow.

$$\frac{\partial \rho}{\partial t} + \nabla \cdot (\rho \mathbf{u}) = 0 \quad (2.12)$$

$$\frac{\partial(\rho \mathbf{u})}{\partial t} + \nabla \cdot (\rho \mathbf{u} \otimes \mathbf{u}) = -\nabla p + \rho \mathbf{g} + \nabla \cdot \mathbf{T} \quad (2.13)$$

$$\frac{\partial(\rho h^t)}{\partial t} + \nabla \cdot (\rho \mathbf{u} h^t) = \frac{\partial p}{\partial t} - \nabla \cdot \mathbf{q} + \nabla \cdot (\mathbf{T} \cdot \mathbf{u}) + \rho \mathbf{g} \cdot \mathbf{u} \quad (2.14)$$

$$\frac{\partial \alpha_k \rho_k}{\partial t} + \nabla \cdot (\alpha_k \rho_k \mathbf{u}) = 0 \quad (2.15)$$

$$\frac{\partial \alpha_1}{\partial t} + \nabla \cdot (\alpha_1 \mathbf{u}) = \alpha_1 \nabla \cdot \mathbf{u} + K_{12} \nabla \cdot \mathbf{u} \quad (2.16)$$

where each phase is represented by one of the $k-s$, and the subscript 1 denotes the liquid. Here ρ is the mixture density field, \mathbf{u} is the mixture velocity vector field, p is the mixture pressure field, T is the viscous stress tensor field, $h^t = h + 1/2\|\mathbf{u}\|^2$ is the mixture specific total enthalpy field, K_{12} is the interface compression coefficient, and \mathbf{q} is the mixture diffusive heat flux [1].

2.7 CLOSURE

The thermodynamic closure proposed by Meng and Colonius is the stiffened gas Equation Of State (EOS) [14]

$$p = (\gamma - 1)\rho\varepsilon - \gamma p_\infty \quad (2.17)$$

The properties of the mixture are the following:

Phase	$\rho [kg\ m^{-3}]$	$c [m\ s^{-1}]$	γ	$p_\infty [Pa]$
Water	997	1450	6.12	3.43×10^8
Air	1.204	343	1.4	0

Table 2.1: Phase properties.

γ and p_∞ are fitting parameters for the stiffened gas EOS [15] derived from the shock *Hugoniot* data [16]. Given that $p_\infty = 0$ for air, the stiffened gas equation becomes the ideal gas law. In the mixing interface region (that is a result of numerical diffusion and not an actual phenomenon), mixture rules have to be precisely defined [17]:

$$\alpha_l + \alpha_g = 1 \quad (2.18)$$

$$\rho = \alpha_l\rho_l + \alpha_g\rho_g \quad (2.19)$$

$$\rho\varepsilon = \alpha_l\rho_l\varepsilon_l + \alpha_g\rho_g\varepsilon_g \quad (2.20)$$

In this case, the perfect fluid equation is employed also for the liquid phase. The system of equations 2.12-2.16 is closed with the following relations:

- the perfect gas equation for the gas phase:

$$p = \rho_2 \frac{R}{M_2 T}$$

- the perfect fluid equation for the liquid phase:

$$\rho_1 = \rho_{0,1} + \frac{1}{R/M_1 T} p$$

- the relation for the interface compression coefficient:

$$K_{12} = \frac{\alpha_1\alpha_2(Z_1 - Z_2)}{\alpha_1 Z_1 + \alpha_2 Z_2}$$

- the Fourier law for the heat flux:

$$\mathbf{q} = -\lambda \nabla T$$

- the Newton and Stokes assumption for the viscous stress tensor:

$$\mathbf{S} = 2\mu \mathbf{E} - \mu \mathbf{I} \nabla \cdot \mathbf{u}$$

where \mathbf{E} is the strain rate tensor and $\mu = \alpha_1 \mu_1 + \alpha_2 \mu_2$ is the mixture dynamic viscosity, M_k are the molar weights of the species involved, $Z_k = \rho_k c_k^2$ are the acoustic impedances calculated using $c_k = \sqrt{\gamma_k R / M_k T}$. The enthalpy of the mixture is calculated as the weighted sum of the enthalpies $h = \sum_{k=1}^2 \alpha_k \rho_k / \rho h_k$ and $\rho_{0,1}$ is the initial density of liquid phase.

3

NUMERICAL INVESTIGATION

3.1 MATHEMATICAL CLASSIFICATION OF FLOWS

Quasi-linear second order PDEs in two variables can be divided into three types: hyperbolic, parabolic and elliptic. This distinction is based on the nature of *characteristics*, curves along which the information about the solution is carried.

In the hyperbolic case, the characteristics are real and distinct. This means that the information propagates in two directions. In general, the propagation is in a particular direction, so that one datum needs to be given at an initial point on each characteristic; the two sets of characteristics therefore require two initial conditions. If there are lateral boundaries, usually one condition is necessary at each point, because one curve is carrying information out of the domain, while the other brings information in.

In parabolic equations the characteristics degenerate to a single real set, meaning that only one initial condition is required. At lateral boundaries, one condition is needed at each point.

In the elliptic case, the characteristics are imaginary and the information propagates in no preferential direction, it travels equally well in all directions. Unsteady problems are never elliptic.

These differences in the equations are reflected in the methods used to solve them. The Navier-Stokes equations are a system of non-linear second order PDEs in four independent variables. Thus, the classification does not directly apply to them. However, they possess some of the properties outlined and many ideas used to solve second order equations in two variables are applicable with some extra care.

3.1.1 *Hyperbolic flows*

Unsteady inviscid compressible flow, which can support sound and shock waves, is essentially hyperbolic. Recalling what already said, supersonic flow only allows informations to propagate downstream. For steady compressible flows, the character essentially depends on the speed. Supersonic flows are described by hyperbolic equations, while subsonic flows are essentially elliptic. Since most of the solution methods are based on the idea that the flow is hyperbolic (and they work quite well), transonic flow is challenging, because two different regimes are described by two different types of equations. Consequently, it is necessary to change the methods for approximating the equations, based on the nature

of the local flow. To make matter worse, the regions cannot be determined prior to the solution.

The equations for viscous compressible flow are even more complicated, and they do not fit very well into the classification scheme. Therefore, numerical methods for them are not very easy to construct.

3.2 COMPUTATIONAL FLUID DYNAMICS

In the very beginning, this field of study was only renowned in high-technology areas, but now, it has rapidly spread across multiple industrial applications and academical research. When applied to supersonic vehicle design and analysis, it plays a crucial role in understanding the complex interactions between the vehicle and its surrounding environment, such as shock waves and airflow.

Nevertheless, computational fluid dynamics has integrated the disciplines of fluid mechanics with mathematics and computer science. Starting from the very bottom, fluid mechanics studies fluids at rest and in motion, whereas CFD mostly focuses on how fluid motion influences heat transfer and also chemical reactions. The characterization of the fluid is described through mathematical equations (*modified* PDEs), converted by computer scientists into a software package. The adjective *computational* refers to using those packages on high-speed computers to obtain the solution of the problem. This method has become the standard for complex interactions and phenomena, as analytical methods require approximations elsewhere anyway. When simulating the interaction between a water droplet and a normal shock wave caused by a vehicle that is flying at supersonic speeds, CFD can provide a detailed understanding of changes in shape, mass distribution, and other critical parameters that can affect the structural integrity of the vehicle. Experiments could be a suitable option in order to capture every physical aspect of the phenomenon without simplifying assumptions. As it is very difficult and cost-intensive, though, being able to numerically model the interaction and obtain accurate results for the distortion and the properties associated with the center of mass is key [5].

3.3 POSSIBILITIES AND LIMITATIONS OF NUMERICAL METHODS

Let us consider a virtual wind tunnel in which a car has to be simulated at a certain speed. In the actual analysis, the car is fixed and air is blown towards it. The floor, though, has to move at the same speed as the wind, which is fairly easy to set up.

On paper, if we accurately solve the three dimensional Navier-Stokes equations (including direct simulation of the turbulence), we obtain a complete data set containing every quantity we need.

The difficulty lies in being able to solve the Navier-Stokes equations **accurately**. Obtaining an acceptably good numerical solution for high Reynolds number is difficult. First of all,

numerical results are always approximate. There are different types of errors that can arise from each part of the process used to obtain the solution:

- Approximation of the mathematical model
- Approximation in the discretization process
- Approximation of the iterative methods used to solve the discretized equations obtained

When the governing equations are known accurately (e.g. the Navier-Stokes equations for incompressible Newtonian fluids), the solution for any desired accuracy can be obtained. However, for many phenomena such as turbulence and multiphase flow, the exact equations are either not available or numerical solution is not feasible.

Even if we solve the equations, though, a comparison with experimental data is needed to validate the numerical model. This usually increases the time needed for the solution and therefore a compromise is usually needed.

In the following chapter, the physical problem is stated and the numerical model used is thoroughly described.

3.4 HYBRID METHOD

The approximation algorithm involves five stages:

1. the application of ACID technique
2. FVM with co-located variables
3. the application of the hybrid Kurganov-Noelle-Petrova (KNP)/Pressure Implicit with Splitting Operators and Semi-Implicit Pressure-Linked Equation (PIMPLE) procedure to approximate convective fluxes
4. the formulation of pressure equation to close the system
5. the approximation of the liquid volume fraction transport equation

3.4.1 VOF method

The already existing VOF method, which was improved by Denner and van Wachem [18] and does not require an explicit interface reconstruction, has been shown to conserve the volume of the fluids. This algorithm it is used to track the interface by the transport of the volume fraction. This method was shown to be more flexible and efficient when dealing with immiscible fluids [19] and performed well when studying the interaction of water droplets with shock waves [20]. The Navier-Stokes equations are solved on averaged fields, whose properties are calculated with respect to the volume fraction. Subsequently, in each computational cell, depending on the local volume fractions, the other fields are

representative of each phase or a mixture of them. In summary, the method is applied to represent the bulk phases using what we call an *indicator function*, typically called *colour function*, with

$$\psi(\mathbf{x}) = \begin{cases} 0 & \text{if } \mathbf{x} \in \Omega_a \\ 1 & \text{if } \mathbf{x} \in \Omega_b \end{cases} \quad (3.1)$$

where $\Omega = \Omega_a \cup \Omega_b$ is the computational domain and the a and b subdomains are occupied by fluid a and b . The indicator function is often called α and will be called that from now on. Among the advantages of the Volume of Fluid (VOF) method, there is the straightforward inclusion of the effect of surface tension.

3.4.2 ACID approach

Recalling that a fluid interface is substantially a discontinuity between densities, while other field variable remain continuous, the jump in density causes an unphysical pressure gradient and acceleration of the flow. Modifying the algorithm is necessary to account for changes in the flow properties at the interface and suppress unwanted oscillatory behavior. The approach of the ACID technique is analogous to the Ghost-Fluid Method (GFM) approach: the interface between phases is considered as a moving internal boundary that requires appropriate conditions. However, when using interface capture methods such as the VOF method, interfacial mesh cells (identified when the volume fraction function is $0 < \alpha < 1$) contain a mixture of phases. This mixture, according to continuum mechanics, has no physical meaning and is the result of a numerical approximation, for it is often referred to as *numerical mixture*. Reconstructing the pressure in those cells is problematic since plausible thermodynamics and hydrodynamic features have to be defined, and the phenomenon simply does not exist in the real world. The ACID technique, developed by Denner et al. and presented in [21], is based on a conservative discretisation of the governing equations in each finite-volume stencil¹. Thus, the color function is kept piecewise constant in each stencil (i.e. imagining that each cell is surrounded by imaginary cells where thermodynamic properties are equal to the central cell), which allows for fully conservative discretisation. In addition, temporal change of the mixture composition is neglected. Convective fluxes become asymmetrical (considering two neighbour cells, the material flux from left to right is not the same as the flux from right to left). However, this approach can be applied directly to balance equations written in differential form, and its conditions can be summarized as follows:

$$\left. \frac{\delta \alpha_1}{\delta x} \right|_f = 0 \quad (3.2)$$

$$\frac{\delta \alpha_1}{\delta t} = 0 \quad (3.3)$$

¹ A finite-volume stencil of a cell P includes all the neighbor cells, as well as P itself

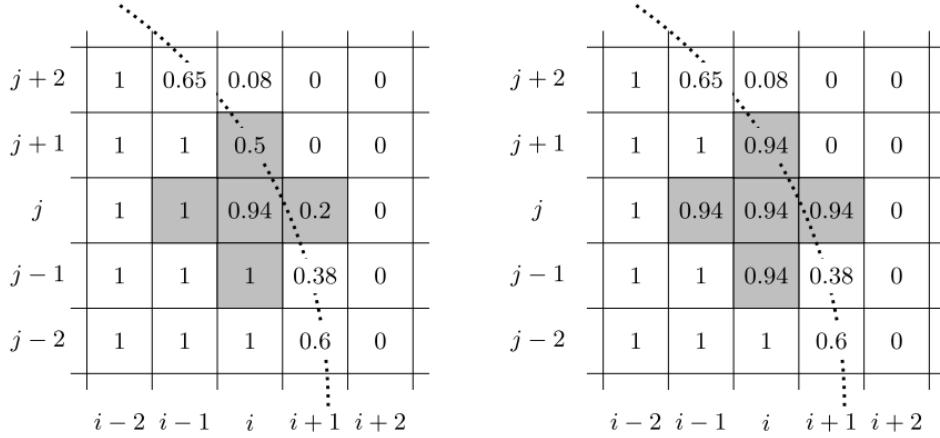


Figure 3.1: Schematic illustration of the colour function $\psi = \alpha$ around cell (i,j) without (left) and with ACID technique (right) [21].

With the application of this scheme, no Riemann solver is needed, as well as no *a priori* assumption on the shock wave has to be made. A conservative approach is required to obtain an accurate description of both the speed and strength of shock waves.

3.4.3 Finite Volume Discretization

Solving the previous set of equations is not easy at all, even for *super* regular domains with simplifying hypotheses on the fluid. Let alone applied for the kind of problems engineers face daily. The conservation equations must then be discretized. For the sake of not overcomplicating the dissertation, the problem is treated in its less complex case: a 1 dimensional advection problem.

$$\mathbf{q}_t + \mathbf{f}(\mathbf{q})_x = \mathbf{0} \quad (3.4)$$

In 3.4, \mathbf{q} is an N-vector of conserved quantities and $\mathbf{f}(\mathbf{q})$ is a nonlinear convection flux. The RHS of the equation can also be non-zero and contain dissipation fluxes, $\mathbf{Q}(\mathbf{q}, \mathbf{q}_x)$, satisfying the weak parabolicity condition: $\nabla_s Q(\mathbf{q}, s) \geq 0$. In the general case, \mathbf{q} is an N-vector that depends on the d spatial coordinates with d corresponding fluxes \mathbf{f} and \mathbf{Q} . The previous equation has to be integrated over the domain $[a, b] \times [t^n, t^{n+1}]$ to obtain the integral form

$$\int_a^b \mathbf{q}(x, t^{n+1}) dx = \int_a^b \mathbf{q}(x, t^n) dx - \int_{t^n}^{t^{n+1}} [(\mathbf{f}(\mathbf{q})(b, t) - \mathbf{f}(\mathbf{q})(a, t))] dt \quad (3.5)$$

Where $\bar{\mathbf{q}}_j^n \approx \frac{1}{\Delta x} \int_{C_j} \mathbf{q}(x, t^n) dx$ is the cell average value over C_j .

The solution is then approximated using a linear reconstruction, using the Taylor expansion of the function \mathbf{q} around $x = x_j$ in C_j :

$$\tilde{\mathbf{q}}^n(x) = \bar{\mathbf{q}}_j^n + (\mathbf{q})_j^n(x - x_j)$$

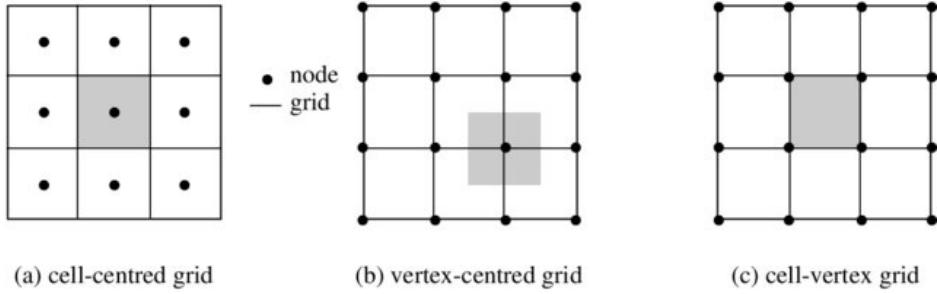


Figure 3.2: Different arrangements of finite volume grids.

The explicit approximation for the discrete mass conservation of the $k - th$ phase under ACID assumption is

$$\frac{\rho_k^{new} - \rho_k^{old}}{\Delta t} + \sum_f Q_{k,f} = 0 \quad (3.6)$$

In order to evaluate the flux integrals, though, one classical approach would be to (at least approximately) solve the Generalized Riemann Problem (GRP)² associated. This problem is so frequent in compressible fluid dynamics that a set of solvers that have been developed go under the name of *Riemann solvers*, and are often called *flux evaluators* by the softwares. Popular examples are:

- **Roe**, based on the linearisation of the Jacobian, which it then solves exactly
- **Harten-Lax-van Leer (HLL)**, the central idea is to assume a wave configuration for the solution that consists of two waves separating three constant states. Assuming that the wave speeds are given by some algorithm, application of the integral form of the conservation laws gives a closed-form, approximate expression for the flux
- **Harten-Lax-van Leer-Contact (HLLC)**, these schemes are more robust but also more numerically diffusive, introduced by Toro, Spruce and Speares [22]

Denner and co-workers [21], instead, pointed out that the HLLC method needs a priori approximations based on the wave conditions and has limited applicability. They proposed the ACID criterion and the corresponding interfacial flow algorithm, which avoids some problems due to the discontinuities and suppresses spurious numerical oscillations. However, the use of the ACID technique described in 3.4.2 introduces an approximation error in the discretized equations of the FVM.

3.4.4 Hybrid approximation of fluxes

Kraposhin et al. [23] used the suggestions of Greenshields et al. [24] and included the semi-discrete central flux formulations together with the face-by-face reconstruction into the pressure-based solver. In this way, the flux formulation and interpolation is stabilized for the investigation of hyperbolic problems. The key to incorporate KNP flux scheme into

² Named after Bernhard Riemann, is a specific initial value problem when the domain has one initial discontinuity

OpenFOAM is understanding its interpolation procedure. The direction of reconstruction is not defined by the surface normal vector as is the case for many central schemes. Left- and right-sided face values are associated with a positive and a negative reconstruction direction. The procedure leads to a pressure equation that allows for the simulation of trans- and supersonic flows with both ideal and real gas effects.

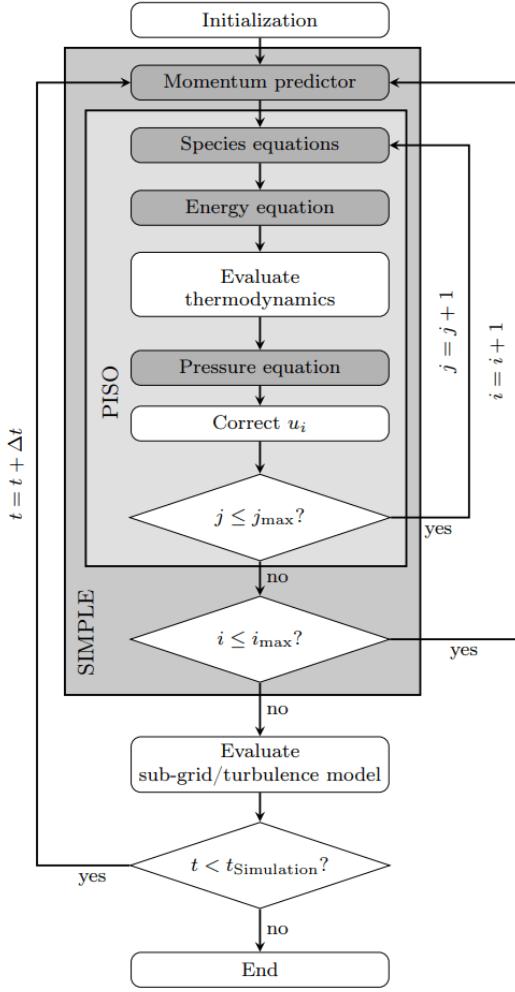
On the other hand, the normal component of the gradient is approximated using linear interpolation. For example, for the heat flux \mathbf{q}

$$(\mathbf{q} \cdot \mathbf{n}) \approx -\lambda_f \frac{\delta T}{\delta \mathbf{n}_f} \quad (3.7)$$

where the operator $\frac{\delta}{\delta \mathbf{n}_f}$ denotes the numerical approximation of face normal derivative. The inherent limitation of the KNP scheme lies in the medium velocity: the minimum Mach number must approach the transonic conditions. On the other hand, many methods currently employed for solving subsonic problems, such as Pressure-Implicit with Splitting of Operators (PISO), Semi-Implicit Method for Pressure Linked Equations (SIMPLE), and their combinations, are also used for high-speed flows with the inconvenience of having numerical oscillations. Thus, to have the best of both worlds, a hybrid scheme was developed, where a blending function is used to decide which approximation of the fluxes grants a better performance. The reader can further investigate the numerical method referring to the cited author.

3.4.5 Pressure equation

The usage of hybrid flux approximation together with ACID allows to employ the PIMPLE method (in the case study it is used with 3 outer correctors) to construct an iterative algorithm for solution of the FVM discretized equations. The method is a combination of the SIMPLE and the PISO method. All these algorithms are iterative solvers but PISO and PIMPLE are both used for transient cases whereas SIMPLE is used for steady-state cases. The number of outer correctors of the PIMPLE algorithm, as mentioned above, defines how many outer iterations to perform, that is, how many times the system of equations are performed before it is forced to move onto the next time step, regardless of whether that time step has converged or not. The number of outer correctors does not mean that it will perform all of these iterations; it will simply iterate until timestep convergence is reached, and move on to the next time step. Another thing to note about this algorithm is that it will solve with relaxation factors until the last outer iteration. In figure 3.4.5 the outline of the PIMPLE solution procedure is illustrated [25]:



- After the initialisation of the simulation, the discretized momentum conservation equation is solved. The solution of the momentum equation gives back a velocity field that does not satisfy the mass conservation. The convection term is taken from the last timestep/iteration.
- The PISO loop starts and the species transport and energy conservation equations are solved. For the robustness of the solver, the coupling between species, energy, and pressure is essential. After the mixture composition and energy are updated, the thermodynamics closure is evaluated.
- The velocity is evaluated at the centroids and interpolated to obtain the values on the CV faces. Then, the pressure equation is constructed and using it ensures updated mass fluxes that satisfy mass conservation

The last step involves evaluating sub-grid/turbulence model.

Usually, the Courant-Friedrichs-Levy (CFL) condition ensures stability

$$CFL = \frac{\|\mathbf{u}\| \Delta t}{\Delta x} \quad (3.8)$$

but in this case, the fully implicit PIMPLE approach is employed where M is small. For large Mach numbers, 3.8 is satisfied automatically if

$$Co^m = V^{-1} \Delta t \sum_f |\Phi_{v,f}| < 1 \quad (3.9)$$

3.4.6 Liquid phase volume fraction transport equation

The discrete transport equation is solved using the Multidimensional Universal Limited Explicit Solver (MULES) limiter, which limits the flux of the variables to guarantee the boundedness and monotonicity of the solution, while keeping the 2nd order approximation

3.4.9. The flux $\Phi_{V,f} = \mathbf{U} \cdot \mathbf{S}$ is calculated according to the continuity equation of the mixture. The equation is obtained by summing phase mass equations normalized by the densities:

$$\nabla \cdot \mathbf{U} = - \sum_k \frac{\alpha_k}{\rho_k} \frac{d\rho_k}{dt} \quad (3.10)$$

Fluxes are calculated using a procedure similar to PISO and SIMPLE:

1. $\Phi_{V,f}$ is split into prediction $\Phi_{V,f}^*$ and correction $\Phi_{V,f}'$
2. prediction is calculated using mass flux of liquid phase $\Phi_{V,f}^* = \Phi_{k,f} / \rho_{k,f}$
3. correction is assumed to be proportional to the gradient of some corrected pressure $\Phi_{V,f}' = -(A^{-1} \nabla p')_f \cdot \mathbf{S}$
4. discrete Poisson equation is formulated for p' :

$$V^{-1} \sum_f \Phi_{V,f}^* - V^{-1} \sum_f (A^{-1} \nabla p')_f \cdot \mathbf{S} = - \sum_k \frac{\alpha_k}{\rho_k} \frac{\delta \rho_k}{\delta t} \quad (3.11)$$

$\frac{\delta \rho_k}{\delta t}$ is the explicit approximation of material derivative for ρ_k

3.4.7 The overall numerical algorithm

The solution is constructed as follows:

1. Variables initialization
2. Compute following time value as $t^{n+1} = t^n + \Delta t$
3. Store variables and fields from the previous time step
4. Predict the density of every phase by solving each continuity equation separately, using values of the mass fluxes from the previous time step
5. Start the PIMPLE loop:
 - a) Update fluid properties based on T and ρ (ψ_k , c_k , Z_k and K_{12})
 - b) Solve the volume fraction transport equations
 - c) Assemble the matrix for the momentum equation and evaluate $H(\mathbf{U})$ and A
 - d) Solve the energy equation
 - e) Update compressibility coefficients ψ_k
 - f) Update speeds of sound c_k
 - g) Update central weights of the KNP scheme
 - h) Update central fields
 - i) Solve the pressure equation

- j) Calculate new values of mass fluxes from the pressure equation
 - k) Update value of the blending KNP / PIMPLE function
 - l) Find phase densities using a corresponding EOS
 - m) Reconstruct velocity with the new pressure gradient
 - n) Update kinetic energy terms $1/2|\mathbf{U}|^2$ and calculate $\partial p/\partial t$ contribution to the energy equation
6. Return to step 2 in case of PIMPLE iterations stopping, otherwise go to step 4

3.4.8 Workflow

The actual workflow will make use of the method previously described, and will be made of 4 distinct phases:

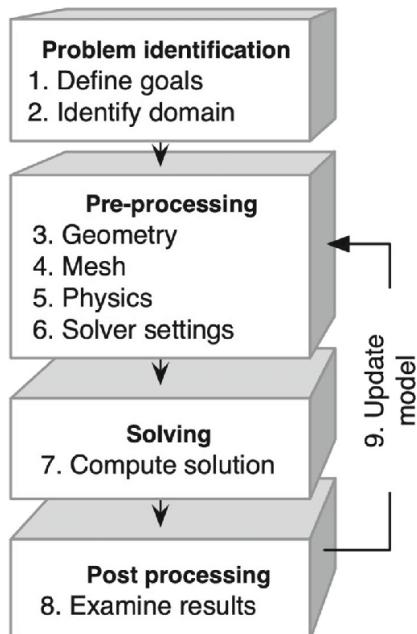


Figure 3.3: Workflow of a typical CFD simulation loop.

The phases of the diagram will be described in the following chapter.

3.4.9 Note: MULES limiter

MULES approach takes the following variables in input: the normal convective flux, and the actual explicit flux of the variables, which is used to return the limited flux of the bounded solution. The normal flux of the variables to be solved are:

$$\Phi_\gamma = [\gamma \bar{\mathbf{U}} + \gamma(1-\gamma) \bar{\mathbf{U}}_{r\gamma}]_f \cdot \mathbf{S} = \gamma [\Phi_u + (1-\gamma)\Phi_{r\gamma}] \quad (3.12)$$

$$\Phi_\beta = [\beta \bar{\mathbf{U}} + \beta(1-\beta) \bar{\mathbf{U}}_{r\beta}]_f \cdot \mathbf{S} = \beta \left[\frac{\Phi_\gamma}{\gamma} + \left(1 - \frac{\beta}{\gamma}\right) \Phi_r \right] \quad (3.13)$$

where $\Phi_u = \bar{\mathbf{U}}_f \cdot \mathbf{S}$ is the volumetric flux through the cell face. Using the normal flux of γ and β , one can determine the mass flux at the cell faces:

$$\rho \Phi = \rho \bar{\mathbf{V}} \cdot \mathbf{S} = \rho_g \Phi_u + (\rho_p - \rho_f) \Phi_\beta + \left[(\rho_m - \rho_g) - \frac{\beta}{\gamma} (\rho_p - \rho_f) \right] \Phi_\gamma \quad (3.14)$$

The MULES loop in OpenFOAM is handled this way:

1. Solve the γ equation with the function `MULES::explicitSolve01(γ , Φ_u , Φ_γ)` that returns the limited flux Φ_γ
2. The limited flux is used to obtain Φ_β and the solution of the β equation, calling the function `MULES::explicitSolve(β , $\Phi_{u\beta}$, Φ_β , β_{min} , β_{max})`

These steps are iterated inside what is called $\gamma - \beta$ -subcycle [26].

4

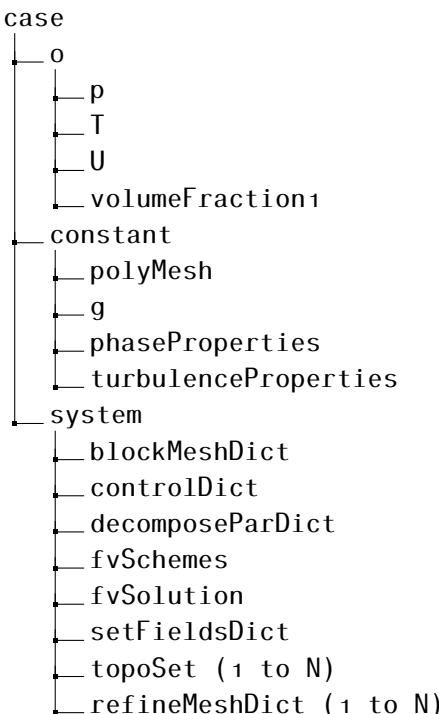
APPLIED CFD

4.1 ABOUT OPENFOAM

OpenFOAM is the free, open source CFD software developed primarily by OpenCFD Ltd since 2004. It has a large user base across most areas of engineering and science, from both commercial and academic organisations. OpenFOAM has an extensive range of features to solve anything from complex fluid flows involving chemical reactions, turbulence and heat transfer, to acoustics, solid mechanics and electromagnetics. The entire library of solvers is written in C++. The package is distributed under General Public License (GPL), allowing the final user to modify and redistribute the software, in both private and commercial ways (An example of commercial redistribution of OpenFOAM is simscale, that also adds a Graphical User Interface (GUI) and offers cloud-based simulations).

4.1.1 *OpenFOAM case structure*

The OpenFOAM case structure is the following:



Each one of the files has a specific purpose in the pre-processing and the analysis part. Specific *dictionaries* can also be added to include certain specific features.

4.1.1.1 0

The *0* folder contains all the initial conditions for the computational domain. It is necessary to initialize all the regions specified in the *polyMesh* folder. Everytime a simulation is started, this folder is generated from another folder called *0.orig*, to restore it to the default when the case is cleaned.

4.1.1.2 constant

This folder contains all the quantities that are constant throughout the simulation. Specifically:

- *polyMesh*: this folder contains all the informations about the mesh created in the pre-processing part, such as the coordinates of all the nodes and some particular zones that can be defined (such as porous media)
- *phaseProperties* contains all the informations about the two phases in the simulation; the equation of state can be specified here
- *turbulenceProperties* lets the user define what turbulence model has to be used as closure; available models range from Reynolds Averaged Navier-Stokes (RANS) to Large Eddy Simulations (LES)

4.1.1.3 system

The *system* directory contains all the dictionaries that the software uses for the solution:

- *blockMeshDict*, the *N topoSet* and the *N refineMeshDict* define the mesh generation workflow, containing all the informations about the base geometry, the discretization process and refinement zones
- *controlDict* contains all the informations about the simulation (runtime, Δt , CFL number, etc.); it is also possible to define custom functions that operate on the data calculated in the simulation
- *decomposeParDict* lets the user divide the mesh among multiple processors and run the simulation in parallel, saving computation time
- *fvSchemes* contains all the discretization schemes, typically being temporal integration method, gradient discretization, laplacian discretization and interpolation method; a wide range of commonly used schemes is available (both 1st order and higher order)
- *fvSolution* lets the user define the solution methods for each variable in the control volume, as well as preconditioner (if available), and non-orthogonal correctors

4.2 PRE-PROCESSING

4.2.1 Mesh generation workflow

4.2.1.1 *blockMesh*

The computational domain is created using the *blockMesh* utility. Defining the domain, as long as it is simple, is fairly straightforward. The coordinates of the bounding box are defined and then connected, creating the bottom and top faces of the domain.

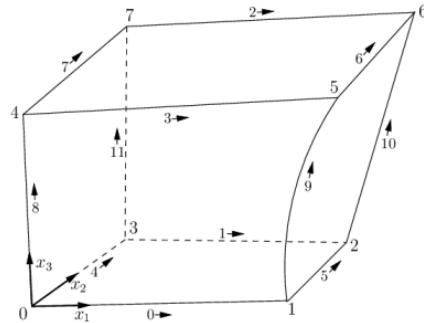


Figure 4.1: A single block.

The exact order must be followed when defining each block. Once the geometry is created, it is possible to set the number of cells in each direction. No bias has been set; the refinement will be handled separately. The unrefined mesh appears as follows:

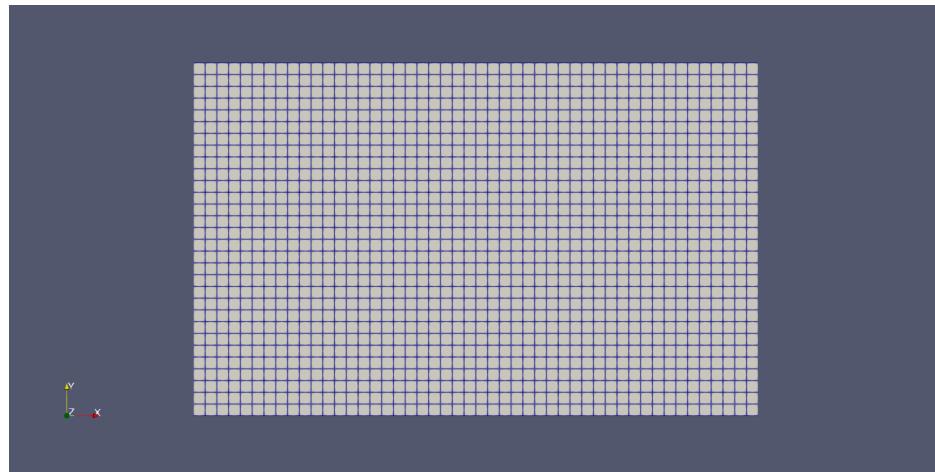


Figure 4.2: Unrefined mesh, 1 CPD.

The refinement loop is explained below:

- **topoSet** is used to create a subset of cells (defined in the *system/topoSetDict*) that will later be refined
- **refineMesh** is used to refine the selected subset of cells. Each original cell is divided in 4 parts, so as each dimension is split in half

The presented loop is reiterated various times, to get a finer mesh closer and closer to the droplet. The scaling factor between the coarsest and the finest cell size, given N refinement loops, is 2^N . The bisection along each direction is used to preserve the stability of the solution.

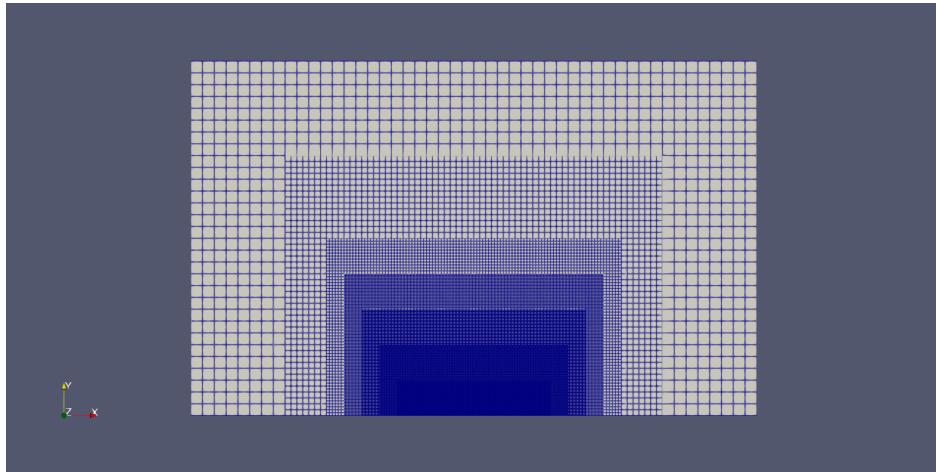


Figure 4.3: Mesh after refinement.

4.2.1.2 *Mesh quality control*

After the creation and the refinement loops are finished, the mesh is checked using the command `checkMesh`. This command initializes a quality control rundown on the created mesh, stored in the `constant/polyMesh` folder. The output is made of four parts:

1. mesh statistics
2. topology check
3. geometry check
4. conclusion

The *mesh statistics* output tells the user how many cells, points, faces there are in the mesh, as well as how many cells are of each type of solid (hex/tet/poly/etc...). The *topology check* tells the user the different patches/boundaries definitions, as well as the point ordering. The *geometry check* outputs a number of quality parameters calculated for the analyzed mesh, including:

- **cell volumes:** the difference between min and max should be as small as possible
- **non-orthogonality:** measures the angle between the line connecting two cell centres and the normal of their common face - 0.0 is the best
- **skewness:** measures the distance between the intersection of the line connecting two cell centres with their common face and the centre of that face - smaller is better

Lastly, the *conclusion* is used to tell if the mesh is usable in a subsequent analysis.

4.2.2 Boundary conditions

As already said in 3.1, the correct set of boundary conditions for a given set of PDEs depends on the equation itself. A quick way to recognize the type of Partial Differential Equation (PDE) is based on the discriminant of the equation. Supposed that $u_{xy} = u_{yx}$, and Lower Order Terms (LOT), a general PDE can be written as follows:

$$Au_{xx} + Bu_{xy} + Cu_{yy} + \text{LOT} = 0 \quad (4.1)$$

The coefficients A, B, C may depend on x and y . If $A^2 + B^2 + C^2 > 1$ in $\Omega \subseteq \mathbb{R}^2$, the equation is 2nd order in Ω . As conic sections and quadratic forms are classified as parabolic, hyperbolic, and elliptic, the classification is based on the discriminant. The discriminant in this case is $\Delta = B^2 - AC$:

- $\Delta < 0 \iff$ Elliptic
- $\Delta = 0 \iff$ Parabolic
- $\Delta > 0 \iff$ Hyperbolic

PDEs can also be classified studying the matrix associated with the linear operator that describes the PDE ¹. The current case study is described by a hyperbolic IBVP. Hyperbolic PDEs retain any discontinuity in the initial variables and / or derivatives.

The standard available boundary conditions include the traditional Dirichlet condition, used to specify the value of the field variable, being *fixedValue*, and a Neumann condition, which allows to set the derivative of the field variable, called *zeroGradient*. The set of boundary conditions is the following:

	inlet	outlet	walls	symmetry faces
p	zeroGradient	zeroGradient	zeroGradient	empty
T	zeroGradient	zeroGradient	advective	empty
U	fixedValue	zeroGradient	Slip	empty
α	fixedValue	zeroGradient	zeroGradient	empty

Table 4.1: Boundary conditions of the IBVP.

4.2.3 Non-Reflective Boundary Conditions

When dealing with pressure waves and propagating temperature disturbances, a particularly useful class of boundary conditions is the NRBC. For an accurate simulation, it is optimal to let the produced pressure and temperature waves flow smoothly out of the domain through the boundary. Reflection at the boundary has an important effect on the

¹ Hyperbolic have only one negative eigenvalue and all the rest are positive, or there is only one positive eigenvalue and all the rest are negative.

solution, especially when we perform a compressible flow simulation, as seen in the following images ²:

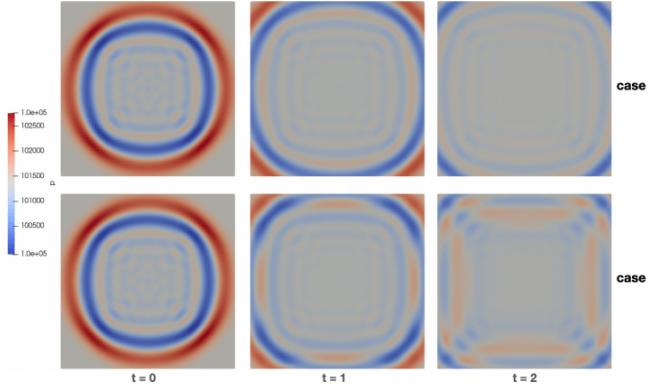


Figure 4.4: Comparison between standard BCs (1) and NRBC (2) [27].

In OpenFoam, we can use two of them, being *advective* and *waveTransmissive*³. They determine the boundary value by solving the following equation:

$$\frac{D\Phi}{Dt} = \frac{\partial\Phi}{\partial t} + \mathbf{U} \cdot \nabla\Phi = 0 \quad (4.2)$$

where $\mathbf{U}(\mathbf{x}, t)$ is the advection speed and $\frac{D\cdot}{Dt}$ is the material derivative. Assuming that the advection velocity is normal to the boundary, we have

$$\frac{D\Phi}{Dt} \simeq \frac{\partial\Phi}{\partial t} + U_n \frac{\partial\Phi}{\partial \mathbf{n}} = 0$$

Those two conditions are different in the way the program evaluates the normal velocity. For the *advective type*, the advection speed is the component of the velocity normal to the boundary:

$$U_n = u_n$$

The *waveTransmissive* type is different. It calculates the advection speed using the thermo-physical properties of the gas:

$$U_n = u_n + c = u_n + \sqrt{\gamma/\psi}$$

where γ is the specific heat ratio (C_p/C_v) and ψ is the compressibility of the gas.

When using non-reflecting boundary conditions the pressure distribution tends to drift towards usually smaller values; this might be due to the implementation in OpenFOAM of the wave equation. Usually, a parameter called *lInfl* is set to make the condition partially non-reflective and mitigate the pressure drift. The solution time could really benefit from a significant decrease in the computational domain, so further analyses will be conducted.

² Courtesy: Leandro Lucchese

³ The solver was updated to allow the correct use of said conditions

	inlet	outlet	walls	symmetry faces
p	fixedValue	zeroGradient	waveTransmissive	empty
T	fixedValue	zeroGradient	advective	empty
U	fixedValue	zeroGradient	waveTransmissive	empty
α	fixedValue	zeroGradient	zeroGradient	empty

Table 4.2: NRBCs proposed.

4.3 SOLVER

4.3.1 *hybridCentralSolvers*

hybridCentralSolvers is a set of OpenFOAM solvers that employ the mixed Rieman/Projection advection scheme for the numerical simulation of compressible flows. The advantage of this hybrid scheme goes from universality of PIMPLE projection method, which allows to use the same framework for different type of flows (different EOS, mesh motion, multicomponent flow) and from non-oscillating flux reconstruction schemes, namely Kurganov-Tadmor. The most suitable solver for the current case is expected to be *vofTwoPhaseCentralFoam* [1, 23], an improved version of *interTwoPhaseCentralFoam*. It uses volumetric fluxes for transport to increase robustness. Its solving algorithm and peculiarities are explained in Section 3.4.

4.4 NUMERICAL SCHEME SETTINGS

The following numerical scheme settings were used:

- convective fluxes were approximated using *Minmod* interpolation, that together with a high resolution scheme, makes the solution Total Variation Diminishing (TVD)
- diffusive fluxes were approximated using Gaussian integration with linear interpolation
- temporal derivative was evaluated using Euler first order scheme
- three outer PIMPLE correctors were employed for each time step
- time step was dynamically adjusted to keep the CFL number equal to 0.15
- system of linear algebraic equations were solved by iterative stabilized biconjugate gradient method, preconditioned with diagonal LU decomposition

4.5 POST-PROCESSING

The post-processing softwares are ParaView and Matlab. ParaView is an open source visualization application developed by Kitware Inc. It is frequently used to visualize Open-

FOAM data, as it can be easily installed on Linux. In this work, ParaView is used to export animation and contours. The quantitative comparison follows another procedure:

- coded postprocess function
- data grabbing
- manipulation in Matlab

4.5.1 Coded drift function

```

1 functions
{
drift
{
    libs      ("libutilityFunctionObjects.so");
6      type      coded;
      name      drift;
      writeControl   writeTime;
      writeInterval  1;
      log         true;
      writeFields   true;

    codeWrite
    #{
const volScalarField& volumeFraction1 = mesh().lookupObject<volScalarField>("volumeFraction1");
16      const volVectorField& U = mesh().lookupObject<volVectorField>("U");

Info<<"Integral calculations begins." << endl;
vector intU = gSum(mesh().V()*volumeFraction1.primitiveField()*U.
primitiveField());
      scalar volWAT = gSum(mesh().V()*volumeFraction1.primitiveField());
Info<<"Integral calculations finished." << endl;

21      scalar V2 = 225.8;
      scalar do = 0.0048;
      scalar Ucfr = intU.x()/volWAT;
      scalar Vcfr = intU.y()/volWAT;

      Info<<"Ucm at " << U.mesh().time().value() << " : " << Ucfr/V2 << endl;
      Info<<"Vcm at " << U.mesh().time().value() << " : " << Vcfr/V2 << endl;
      #};
31    }
}

```

This function calculates the following quantities from the solution fields (grabbed with *lookupObject*):

$$\int_{\Omega} \alpha_1 \mathbf{U} d\Omega \approx \sum_{cell_i}^N \alpha_{1,i} \mathbf{U}_i V_{C,i} \quad (4.3)$$

$$V_1 = \int_{\Omega} \alpha_1 d\Omega \simeq \sum_{cell_i}^N \alpha_{1,i} V_{C,i} \quad (4.4)$$

to calculate both component of the center of mass velocity

$$u = \frac{\int_{\Omega} \alpha_1 \mathbf{U} \cdot \hat{\mathbf{x}} d\Omega}{\int_{\Omega} \alpha_1 d\Omega} \simeq \frac{\sum_{cell_i}^N \alpha_{1,i} u_i V_{C,i}}{\sum_{cell_i}^N \alpha_{1,i} V_{C,i}} \quad (4.5)$$

$$v = \frac{\int_{\Omega} \alpha_1 \mathbf{U} \cdot \hat{\mathbf{y}} d\Omega}{\int_{\Omega} \alpha_1 d\Omega} \simeq \frac{\sum_{cell_i}^N \alpha_{1,i} v_i V_{C,i}}{\sum_{cell_i}^N \alpha_{1,i} V_{C,i}} \quad (4.6)$$

The gathered data is then exported into a .xlsx file and used for comparison with the previous literature. For each step, the change in position of the center of mass is evaluated using the velocity at each time step with the trapezoidal integration method

$$y_n = y_{n-1} + (t_n - t_{n-1}) \frac{(u_{n-1} + u_n)}{2} \quad (4.7)$$

The result is virtually the same as using the forward Euler method

$$y_n = y_{n-1} + (t_n - t_{n-1}) u_n \quad (4.8)$$

or the backwards Euler method

$$y_n = y_{n-1} + (t_n - t_{n-1}) u_{n-1} \quad (4.9)$$

In summary, the following software are used:

Process	Tool	Version
Mesh	blockMesh	v2.3.0
Solution	vofTwoPhaseCentralFoam	OF+2312
Postprocess I	ParaView	v5.13.0-RC1
Postprocess II	Matlab	R2024a

Table 4.3: Used softwares.

5

RESULTS

5.1 VALIDATION

5.1.1 Moving contact - 1D Riemann Problem

This test verifies the continuous behavior of pressure, velocity, and temperature near the constantly moving contact discontinuity, which imitates a phase interface. The boundary conditions are presented below:

	sides	symmetries
p, T, α	zeroGradient	empty
U	fixedValue	empty

Table 5.1: Boundary conditions.

Then a region of the domain is initialized with a different α . Here are the results compared with the exact solution:

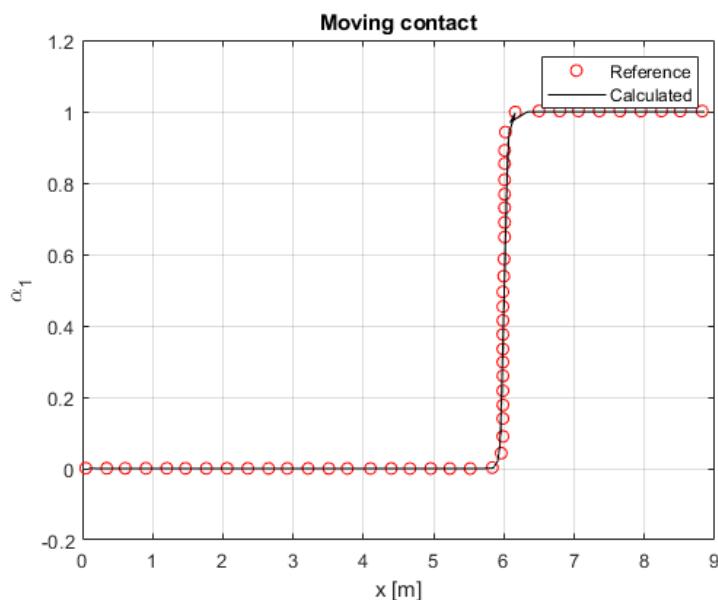


Figure 5.1: Moving contact discontinuity: volume fraction distribution.

5.1.2 Shock tube

The setup with no column of water is identical to *Sod's shock tube* problem. The known analytical solution makes it perfect to be a standard test case for compressible solvers. The robustness of this model and the accuracy of the transient solution were assessed through the comparison with the aforementioned analytical discontinuous solution [28, 29]

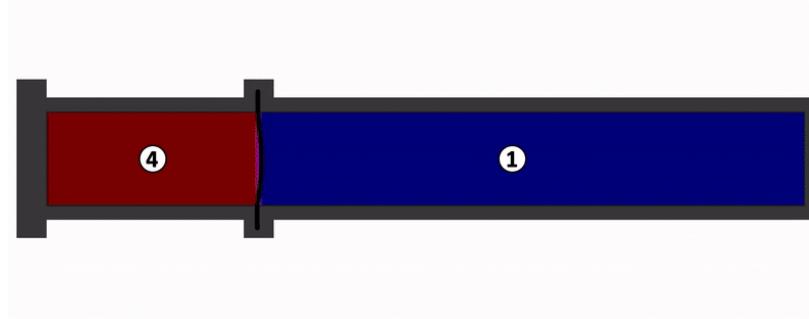


Figure 5.2: Shock tube setup. *Driver (4)* and *driven section (1)*.

$$\frac{p_4}{p_1} = \frac{p_2}{p_1} \left\{ 1 - \frac{(\gamma_4 - 1)(a_1/a_4)(p_2/p_1 - 1)}{\sqrt{2\gamma_1[2\gamma_1 + (\gamma_1 + 1)(p_2/p_1 - 1)]}} \right\}^{\frac{-2\gamma_4}{\gamma_4 - 1}} \quad (5.1)$$

Equation 5.1 gives the incident shock strength p_2/p_1 as an implicit function of the diaphragm pressure ratio p_4/p_1 . Given $M = 1.47$, it is possible to analitically compute p_2/p_1 and numerically solve equation 5.1 to obtain the diaphragm pressure ratio, as shown in 5.3.

$$\frac{p_2}{p_1} = 2.35, \quad \frac{\rho_2}{\rho_1} = 1.81, \quad M_2 = 0.711 \quad (5.2)$$

The intersection with p_2/p_1 gives $p_4/p_1 = 6.32$, resulting in a pre-diaphragm pressure of $p_4 = 6.4 \times 10^5 Pa$ and a density of $\rho_4 = 2.18 kg/m^3$.

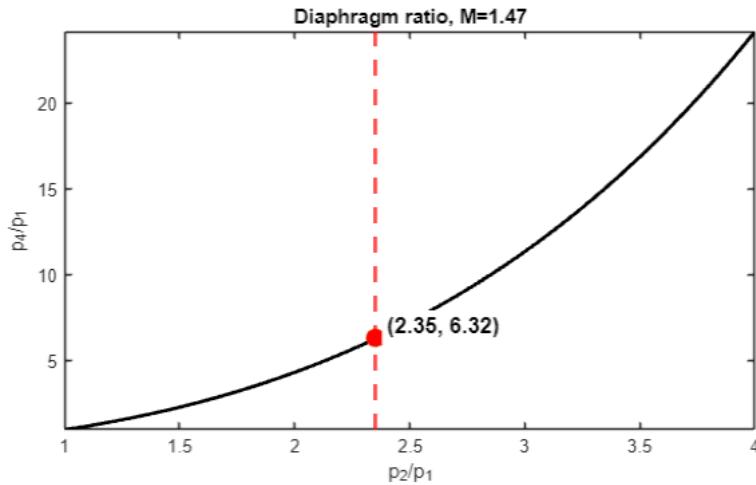


Figure 5.3: Diaphragm ratio and pressure shock ratio for $M = 1.47$.

The mesh is similar to what is shown in figure 4.2, but the horizontal dimension is extended, to allow for a proper shock/expansion development:

	inlet	outlet	walls	symmetry faces
p	zeroGradient	zeroGradient	zeroGradient	empty
T	zeroGradient	zeroGradient	zeroGradient	empty
U	fixedValue	zeroGradient	Slip	empty
α	fixedValue	zeroGradient	zeroGradient	empty

Table 5.2: Boundary conditions for the Shock tube problem.

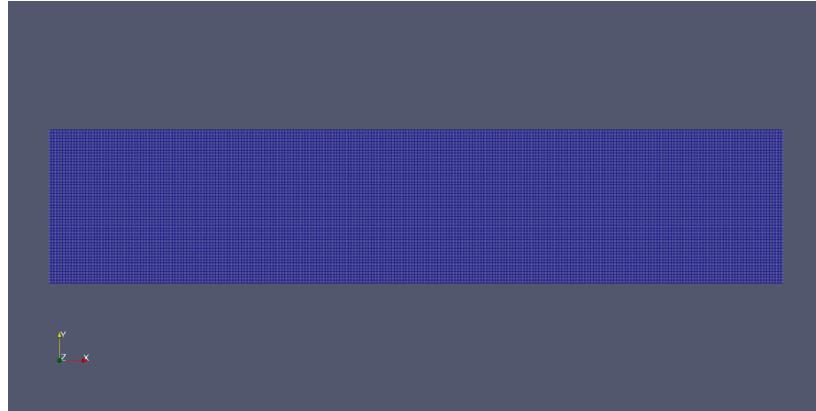
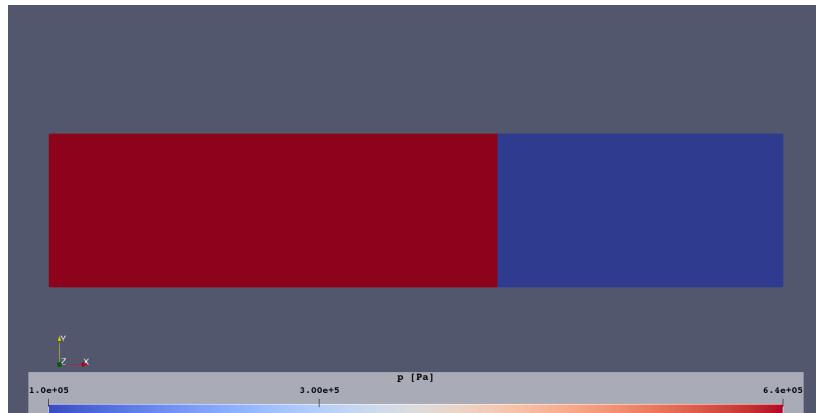


Figure 5.4: Mesh of the shock tube problem, resolution: 16 CPD.

Here, even though the water column is not present yet, it is useful to measure the distances as multiples of the initial diameter d_0 . The driver region, that measures $12d_0$, is initialized with the conditions labelled as 4. The virtual diaphragm is placed at $x/d_0 = -2$, and the results are examined at

$$\bar{t} = \frac{2d_0}{V_S} \sim 1.9 \times 10^{-5} s \quad (5.3)$$

where V_S is the velocity of the shock front.

Figure 5.5: Contour of pressure at $t = 0s$, driver (red) and driven region (blue).

The figures below show 3 different sets of results, as well as a convergence study. As labelled, the **blue** dataset is obtained with a constant resolution of 8 CPD, the **red** is obtained at 16 CPD and the **green** is obtained at 32 CPD. Since the error between the last two cases

is negligible and in the real case the edges of the solution are smoothed by the viscous effects.

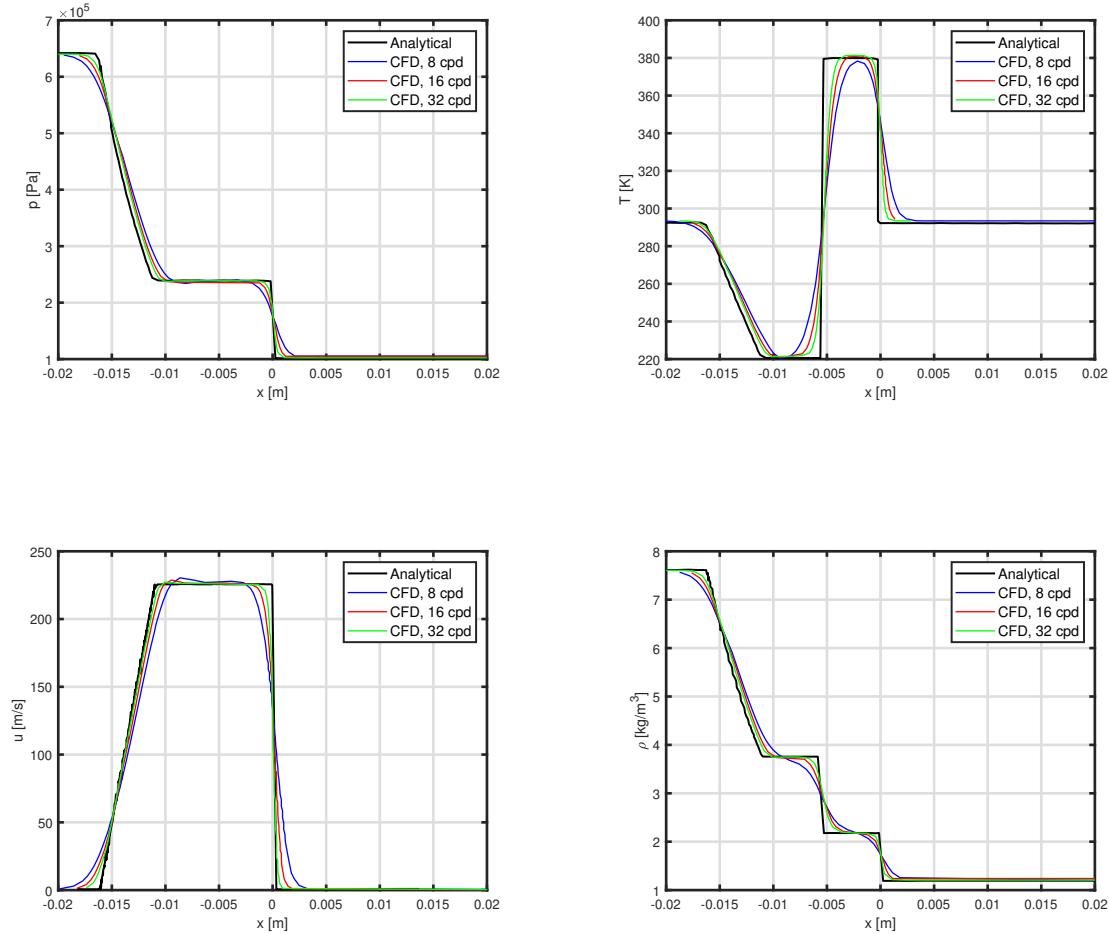


Figure 5.6: Mesh independence study and comparison with theoretical results.

5.1.3 Experimental comparison

The last validation case is the experiment conducted by Igra and Takayama [30], whose setup is presented in the figure below ¹:

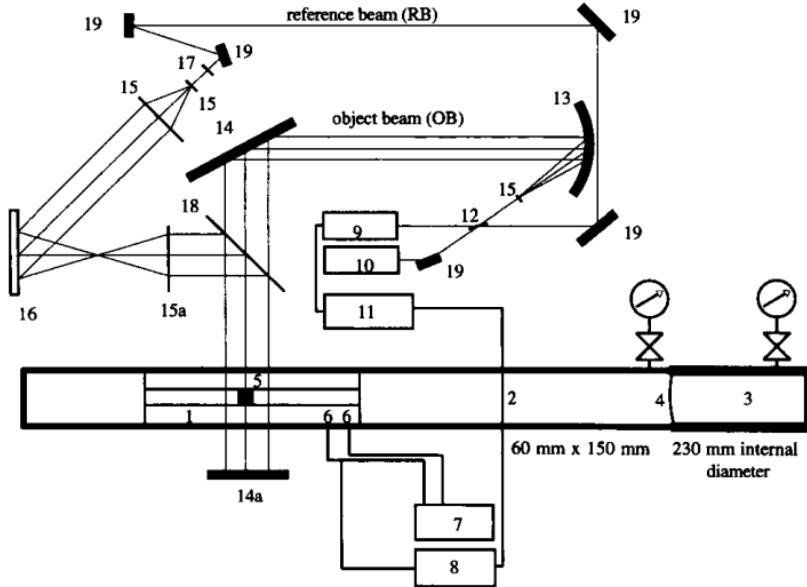


Figure 5.7: Experimental setup.

The shock tube and its optical arrangement are shown. Its low-pressure channel had a cross section of $60 \times 150\text{mm}$ and was approximately 5m long. It was made of smoothly finished 25mm thick steel plates. The high-pressure chamber consisted of 230mm diameter and 1.5m long thick-wall steel tubing. A smooth transition section was inserted downstream of the diaphragm section. The test gas was atmospheric air and the driver gas was nitrogen. The shock tube was arranged so as to place its 150mm wide side wall in a horizontal position. A cookie cutter was inserted into the test section, by which the height of the test section was narrowed from 60mm to 4mm , so that a 4.8mm diameter water column was readily produced at the center of the test section. The incident shock wave Mach number in the present experiments was 1.47 in air.

As of the numerical study, first, several different cases were run to investigate the relevance of the distance between the diaphragm and the droplet surface. The results obtained have shown the significance of that distance, showing that the computational domain cannot be reduced too much.

The experiment has been conducted to measure quantities of interest such as the *leading edge drift* and the *deformation in the flow direction*. The mesh is very similar to 4.3, but given the previously mentioned results on the distance between the cylinder and the wave, it is extended to initialize the high pressure zone further from the droplet. Figures 5.8, 5.9, 5.10, 5.11 show the initialization of the fields at $t = 0\text{s}$ and the different resolutions for the sensitivity study.

¹ With permission from Begell House Inc., Publishers

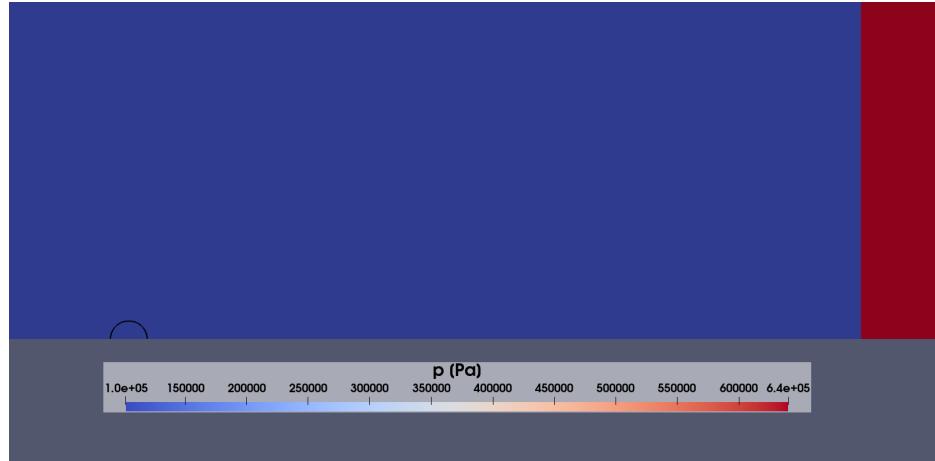


Figure 5.8: Details of the simulation setup at $t = 0s$; iso-alphas are shown in black for $0.0001 < \alpha < 0.99$.

Three different test cases are prepared to study the dependence of the results on the mesh:

- **Mesh I**, 32 CPD, 5 refinement loops
- **Mesh II**, 64 CPD, 6 refinement loops
- **Mesh III**, 128 CPD, 7 refinement loops

The three meshes are shown below²; the final test case has around 2.45M FV cells.

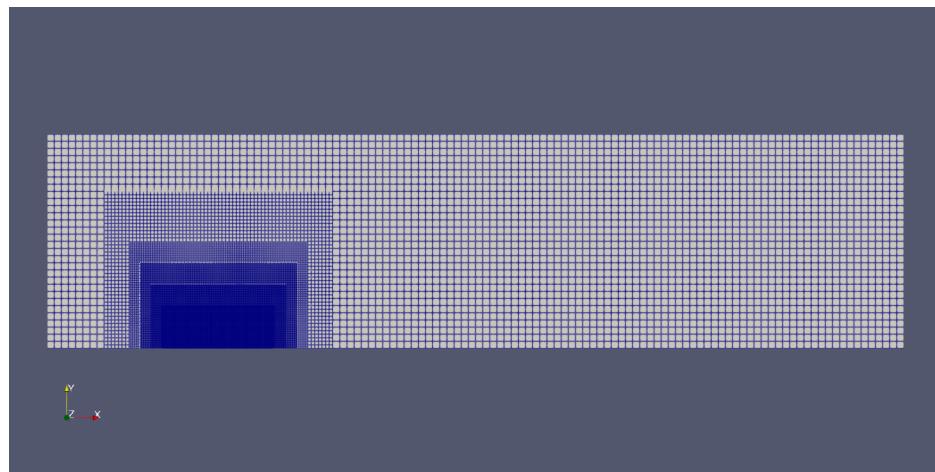


Figure 5.9: Mesh I

² Only half of the mesh is shown for better clarity

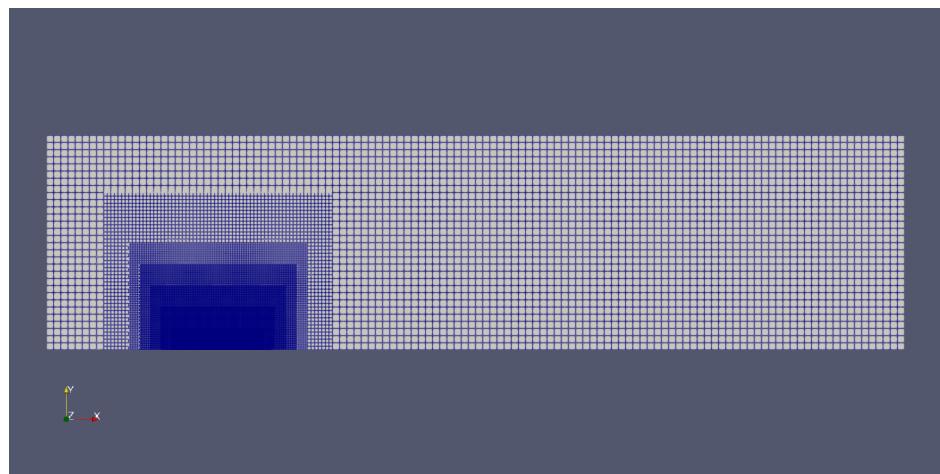


Figure 5.10: Mesh II

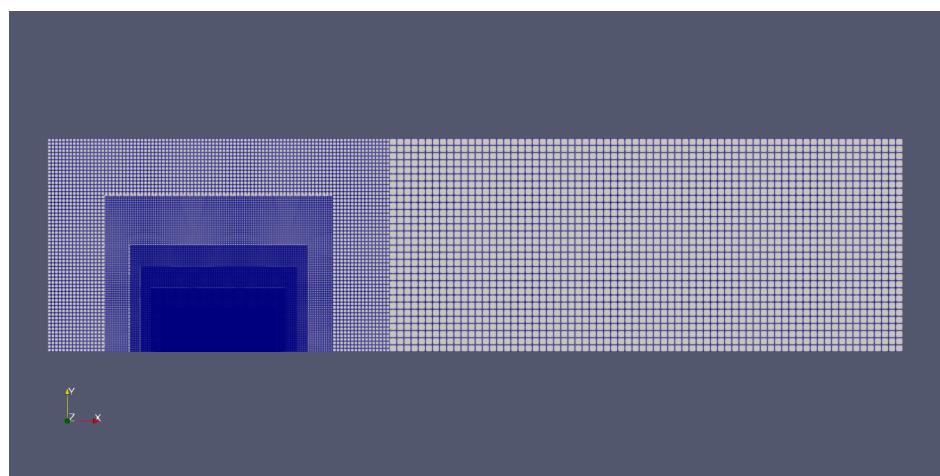


Figure 5.11: Mesh III

All cases were run with $CFL = 0.15$. The distance between the droplet and the initial disturbance is $40D$. Here are the comparison between the three cases and the experimental results:

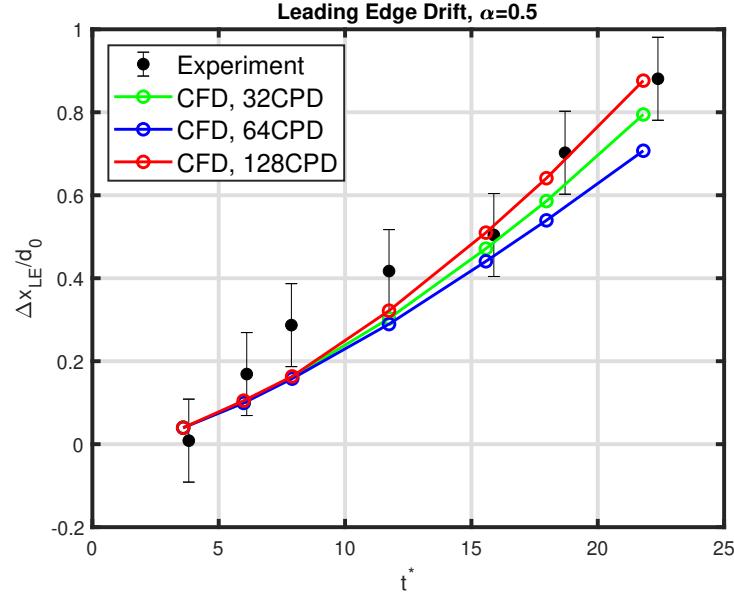


Figure 5.12: Leading edge drift normalized by the initial diameter.

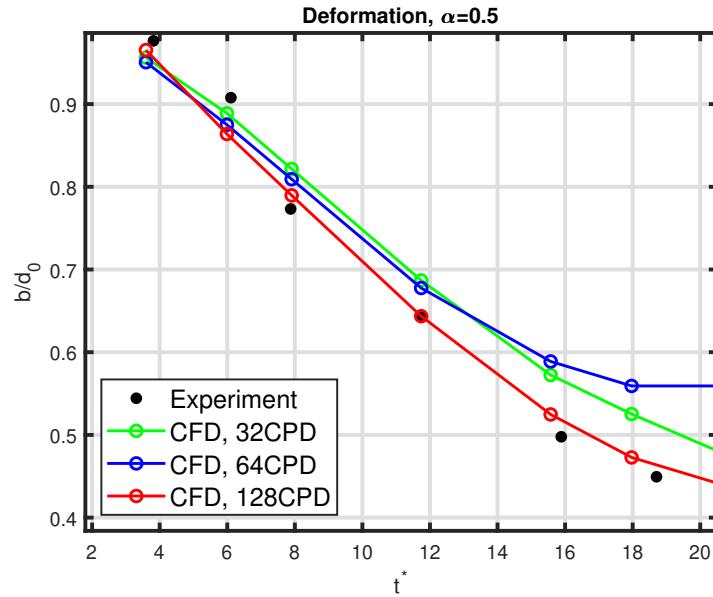


Figure 5.13: Spanwise deformation normalized by the initial diameter.

The numerical results are also compared to the original Igra and Takayama double exposure holographic interferograms³. The numerical contours are obtained by computing the numerical Schlieren function, following those steps with the built-in python calculator in Paraview:

1. Compute the magnitude of the density gradient:

$$|\nabla \rho|$$

2. Normalize it by the maximum value of the field to get an array bounded between 0 and 1:

$$\frac{|\nabla \rho|}{\max\{|\nabla \rho|\}}$$

3. Sharpen the lines applying the following transformation:

$$\Phi = e^{-k \frac{|\nabla \rho|}{\max\{|\nabla \rho|\}}}$$

The process works by allowing normally unobservable changes in a fluid's refractive index to be seen:

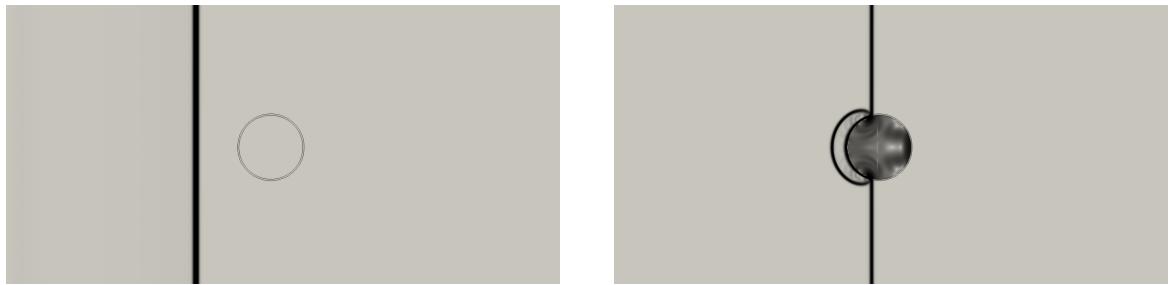


Figure 5.14: Impact of the shock wave on the surface of the cylinder, $0.001 < \alpha < 0.9$.

5.1.3.1 Interaction mechanism

The following pictures (Figg. 5.15, 5.16), are useful in understanding the mechanism of shock wave-column interaction [31]. The incident shock initially creates a reflected wave, then a transmitted and a diffracted wave. Depending on the acoustic impedance of the media, the reflected wave can be a shock wave or an expansion wave. In this case, since $Z_{water} > Z_{air}$, the reflected wave will be a shock wave. The reference time is measured starting from the time of the impact and non-dimensionalized using the inertial time scale

$$t^* = \frac{V_2}{D_0} (t - t_0) \quad (5.4)$$

³ With permission from Springer Nature

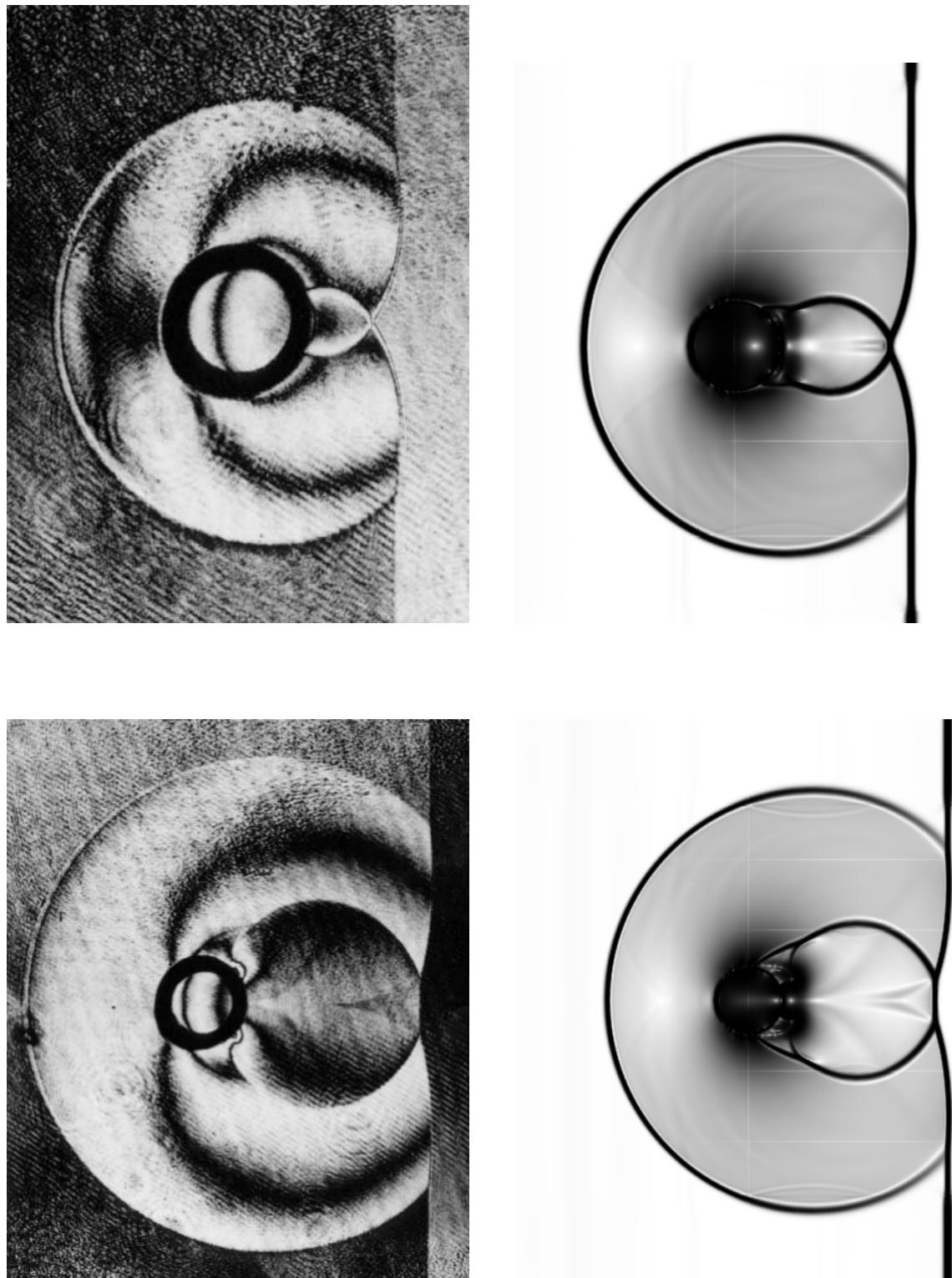


Figure 5.15: Double exposure holographic interferogram from the experiment on the left side ($t_1 = 23\mu s$, $t_2 = 43\mu s$), numerical Schlieren on the right.

Isopycnics (surfaces of constant density) of the gas phase are shown below and compared to the numerical results by Igra and Takayama:

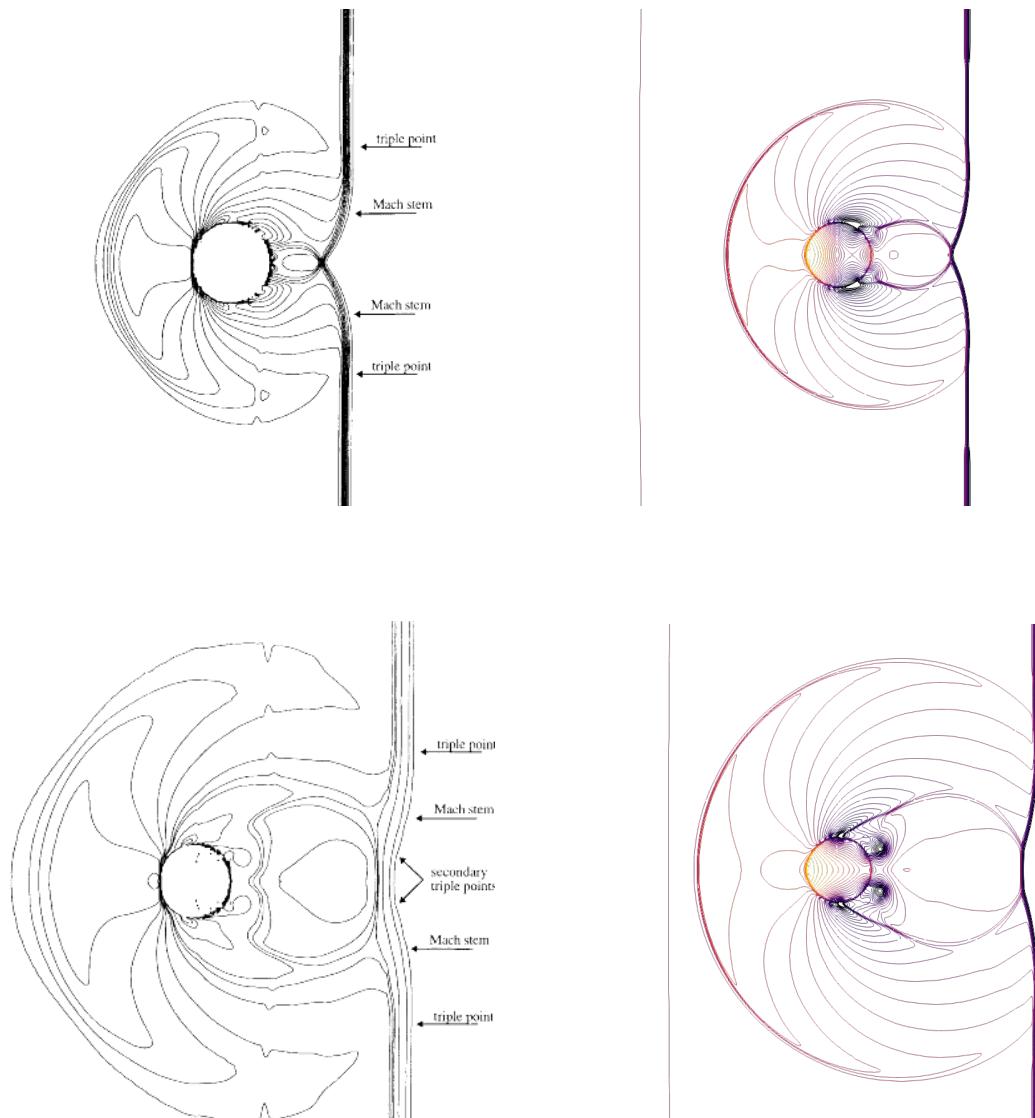


Figure 5.16: Gas phase isopycnics by Igra and Takayama (left, $t_1 = 23\mu s$, $t_2 = 43\mu s$) and computed (right).

Because pressure waves can travel faster in water compared to air, the transmitted wave reaches the leeward side of the droplet earlier. When it reaches the interface, since $Z_{air} < Z_{water}$, the transmitted wave reflects as an expansion wave. When the reflected expansion waves merge with the original shock wave, two *Mach stems* are created. A triple point is formed at the intersection of the incident wave, reflected wave and Mach stem.

The two counteracting flows (the downstream flow from the front stagnation point and the upstream flow from the rear stagnation point) interact, causing the external airflow to develop a toroidal primary vortex near the flow separation point that moves downstream later and forms a recirculation region. At this stage, the external airflow is similar to the flow around a solid sphere of similar Re . Two counter-rotating secondary vortices are also formed between the separation and the rear recirculating flow region.

The combined effects of the recirculating region and the spread of high pressure cause the droplet to deform into a *cupcake shape*. The droplet deformation is further investigated below.

5.1.3.2 Droplet deformation

As a consequence of exposure to the post-shock high-speed flow, after the reaction time has elapsed, the deformation of the droplet can be observed. During the early stages of the deformation, due to the pressure gradient between the two sides of the cylinder, the water phase is flattened in the streamwise direction. The time evolution is illustrated in the figures below; to correctly visualize the deformation and scale, the size and positioning of the water body in the pictures is the same. The peripheral part of the water column is continuously eroded, with sub-droplets being stripped from the original coherent phase and distributed widely in the wake region. These pictures confirm the qualitative features of the so-called shear-stripping breakup mechanism, which is typical at higher We numbers, under the SIE regime. To further analyze the solution, the vortical structures are visualized by plotting the magnitude of the vorticity vector field on the $z = 0$ plane. The contour in the lower part of the figures includes three iso-alpha curves showing the interface between the two phases, at $\alpha_1 = 0.001$, $\alpha_1 = 0.5$ and $\alpha_1 = 0.99$.

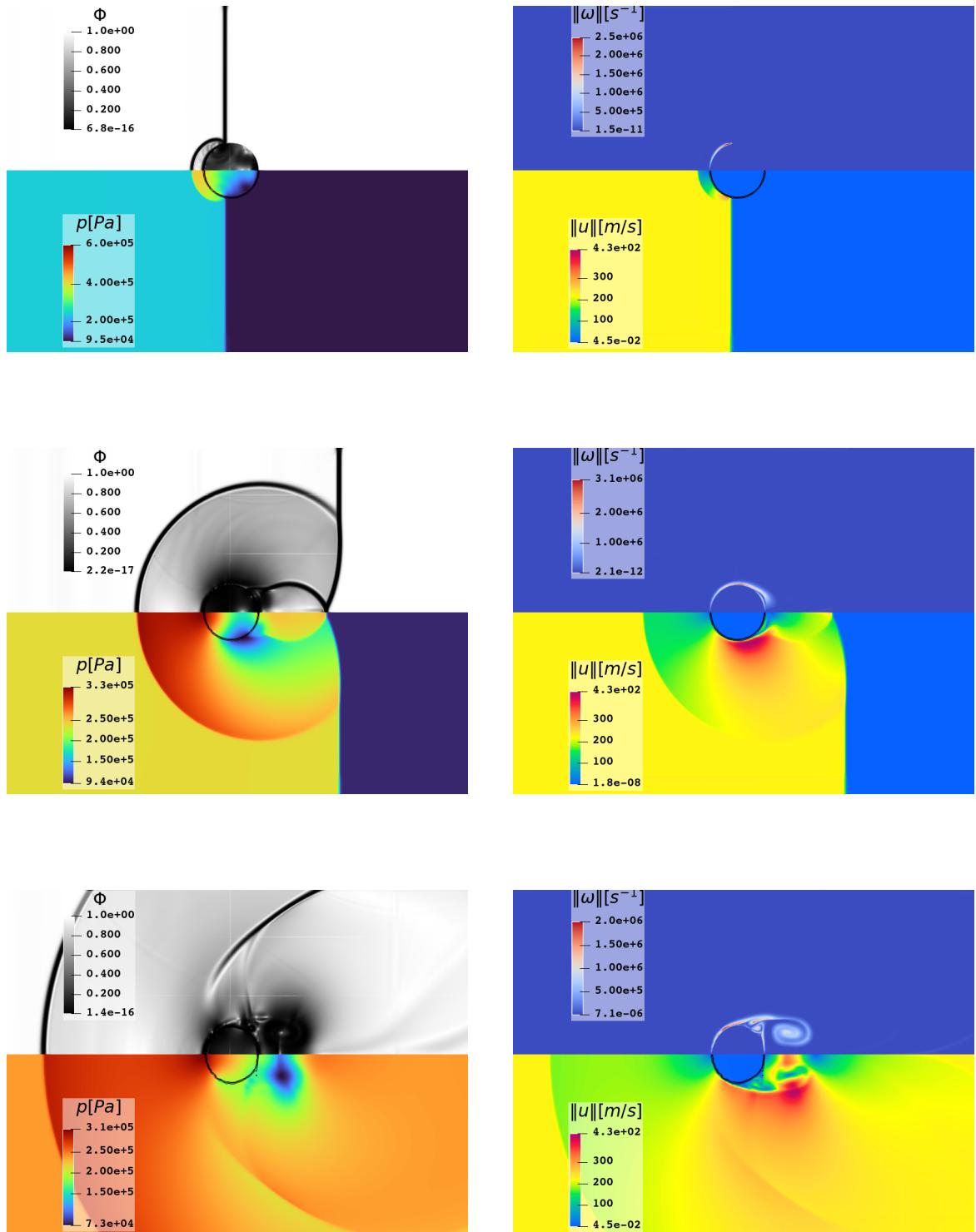


Figure 5.17: Pressure and Schlieren (left); velocity and vorticity magnitude (right), from top to bottom, at $t_1^* = 0.235, t_2^* = 1.176, t_3^* = 3.057$.

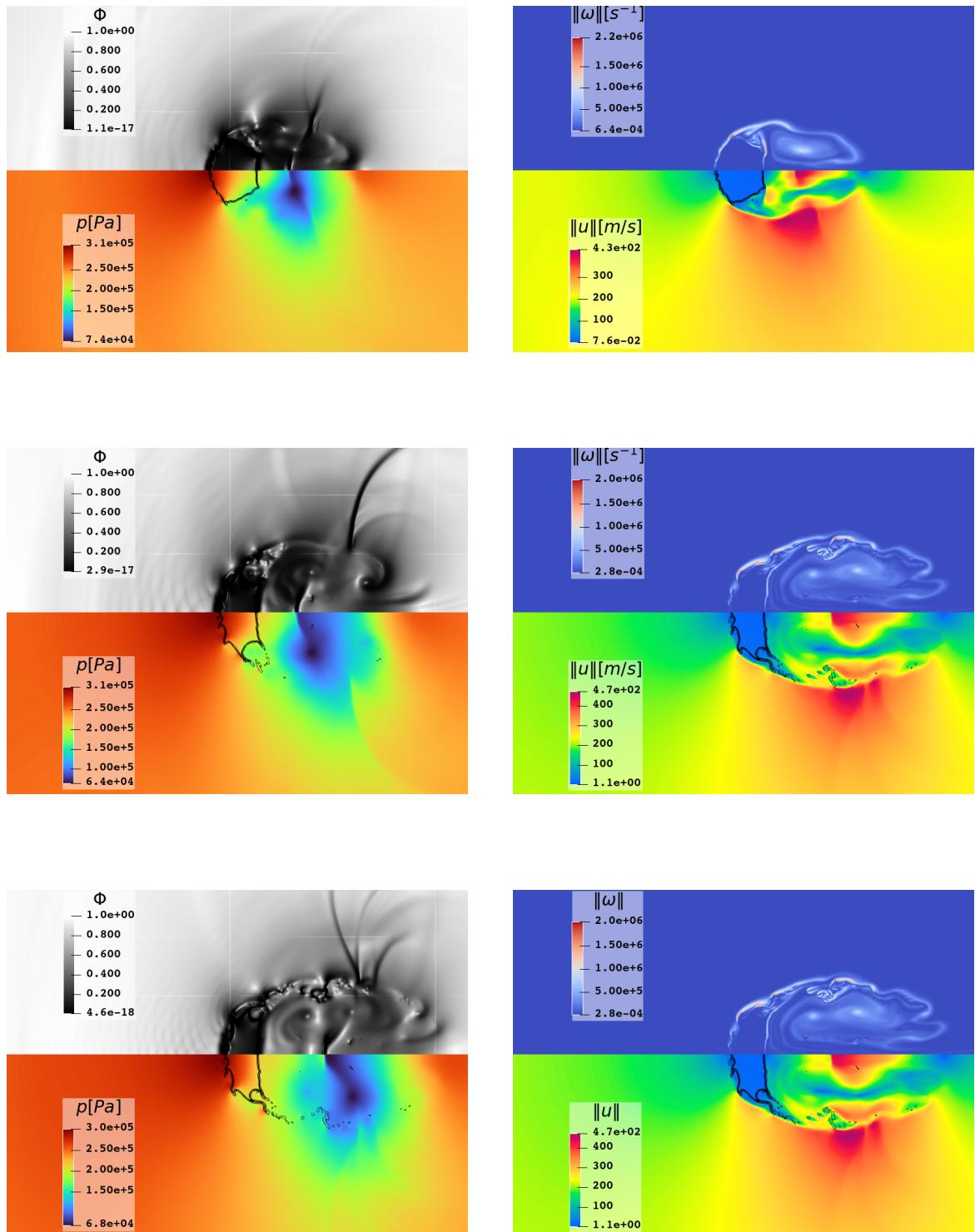


Figure 5.18: Contours of pressure and Schlieren (left) and velocity/vorticity magnitude (right), from top to bottom, at $t_1^* = 5.88$, $t_2^* = 10.584$, $t_3^* = 12.936$.

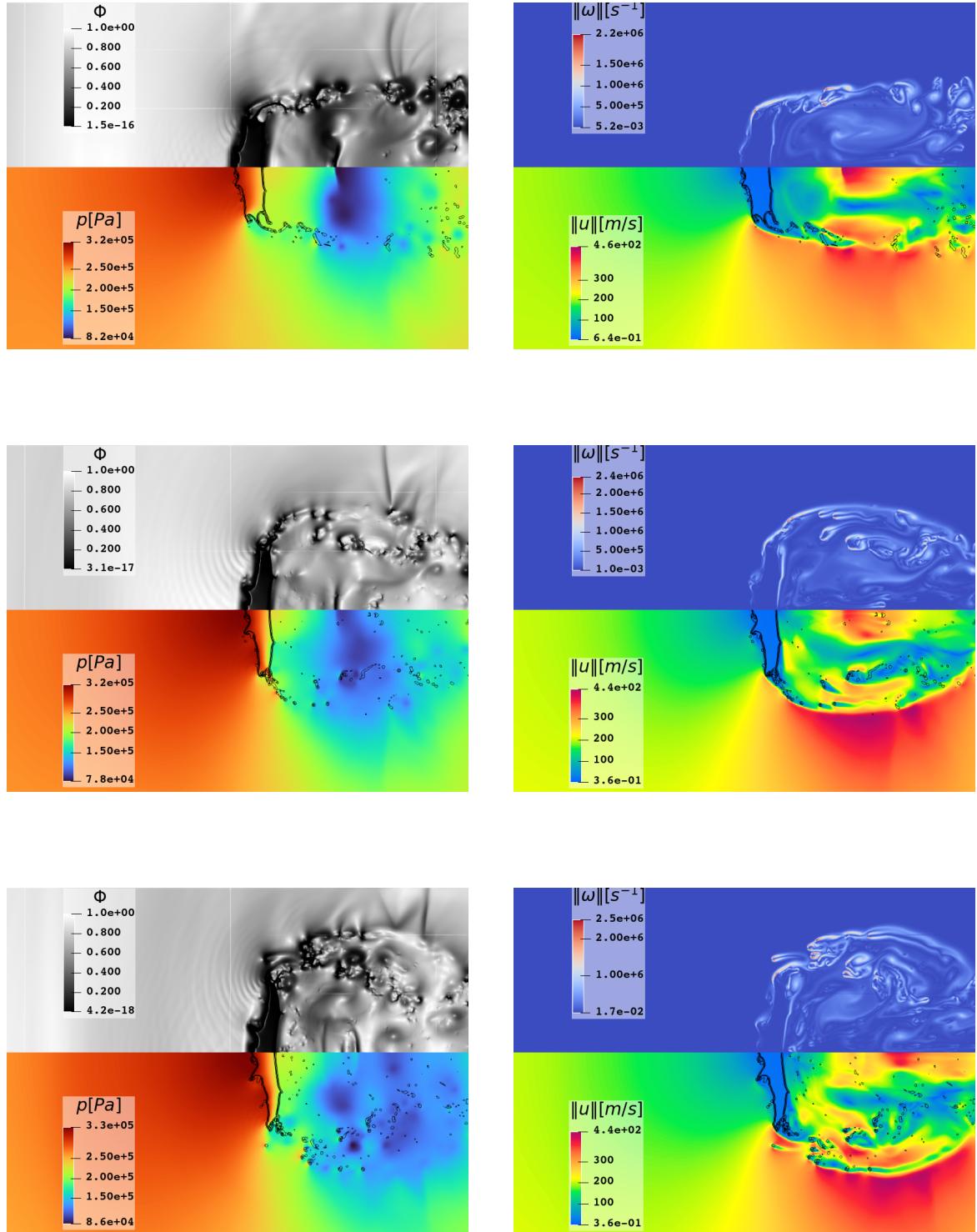


Figure 5.19: Contours of pressure and Schlieren (left) and velocity/vorticity magnitude (right), from top to bottom, at $t_1^* = 16.229, t_2^* = 19.522, t_3^* = 21.404$.

5.2 NUMERICAL COMPARISON

Another part of the work has been dedicated to compare the numerical results with the article by Meng and Colonius [14]⁴, who studied the phenomenon and developed an in-house solver to match the experimental data. They compare the usual quantities of interest to Igra and Takayama, as well as the computation of the drift of the center of mass and its velocity, as shown in subsection 4.5.1:

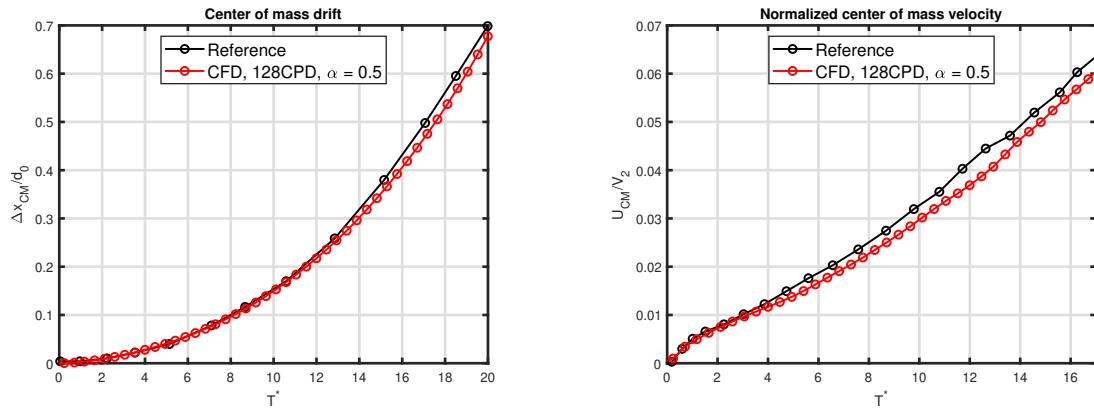


Figure 5.20: Comparison with Meng and Colonius numerical results.

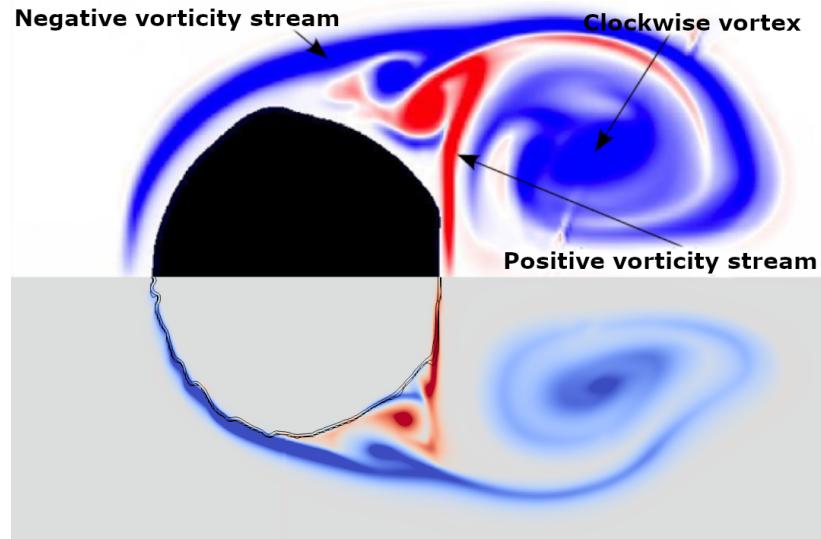


Figure 5.21: Positive and negative vorticity streams interacting to form a recirculation region; reference on top and computed results on the bottom.

⁴ With permission from Springer Nature

Vortices formation can also be studied from the velocity vector plot. KH instabilities start to manifest on the surface of the cylinder and can be observed. Sharma and co-workers studied this phenomenon in [31] and have shown how at lower Weber numbers the formation of KH waves is midway between the front stagnation point and the peripheral zone. Increasing the Weber number makes the waves formation shift towards the stagnation point.

Additionally, when the ratio between gas and liquid density is $\mathcal{O}(10^{-3})$ (such as in the present case studied) the baroclinic effect⁵ is very important and RT instability is the prevalent mechanism for lobe formation because of the high density gradient at the interface.

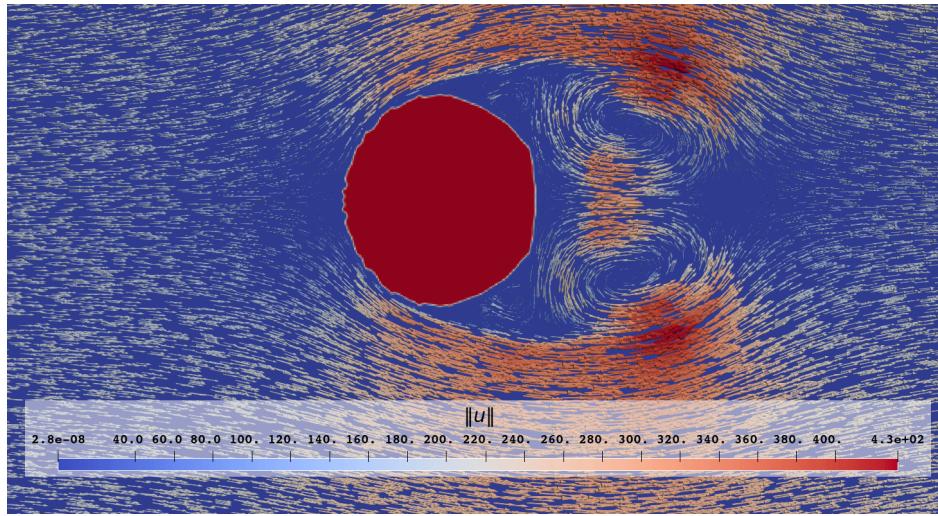


Figure 5.22: Velocity vectors and vortices on the *edges* of the cylinder.

Furthermore, lobes formation undergo the following phases:

- lobe stretching and ligaments formation
- corrugation of the lobes and multiple sub-lobes formation
- formation of holes and detachment of the lobes leading to bridge breakup

Lobe elongation, which results in ligament formation, was demonstrated to be a primary mechanism under conditions of low gas–liquid density ratio. The phenomenon can be observed in Figures 5.18, 5.19 and 5.23.

⁵ The term *baroclinic* describes the process responsible for the creation of vorticity. Typically, the development of vorticity is influenced by factors such as advection, stretching, and twisting, and baroclinic generation. This baroclinic vorticity generation takes place when there is a variation in density along pressure constant surfaces

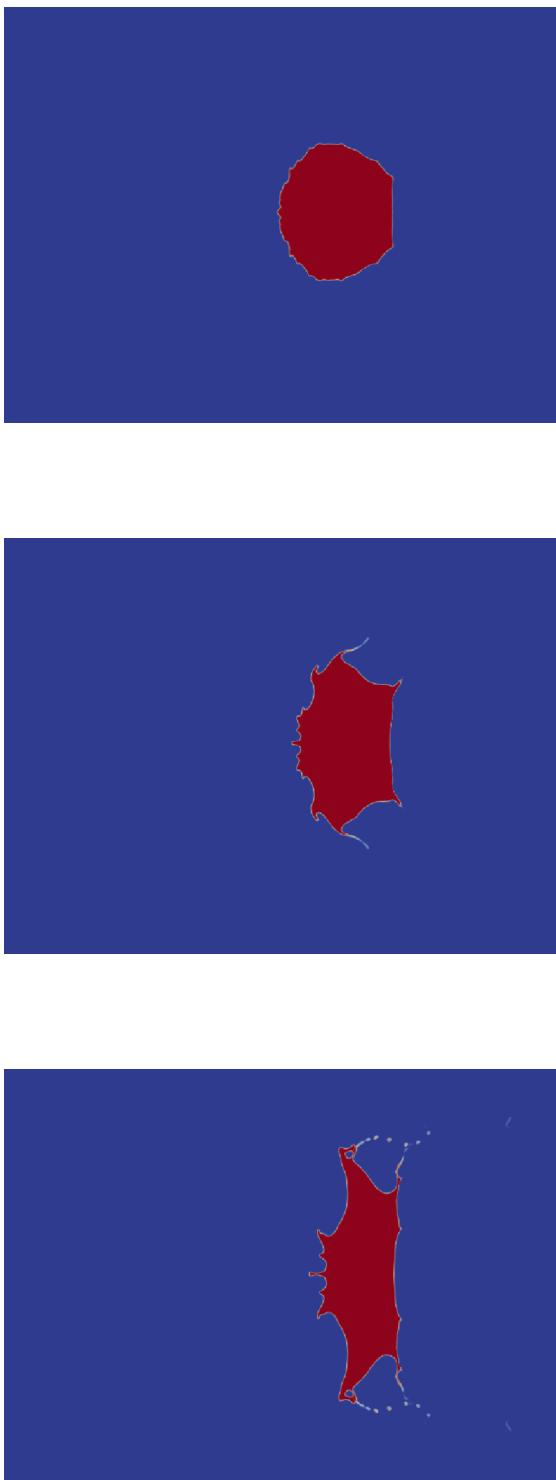


Figure 5.23: From top to bottom: KH instabilities (1), ligaments and lips formation (2) and ligaments breakup (3); contour of liquid volume fraction α from 1 (red) to 0 (blue).

5.3 RESULTS

After the validation process, a different scenario has been examined: two columns of water, placed horizontally on the $x = 0$ line and distant 30mm one from another, are affected by the normal shock. It is needed to say that the analysis is affected by an inherent ambiguity in the criteria used to delineate the boundary of the deforming body, mainly derived from experimental observations. Moreover, Igra and Takayama's discussion does not elucidate these criteria and does not provide error quantification for their deformation measurements, thus, a range of plots is been compared to the experimental data, with $0.25 < \alpha < 0.99$.

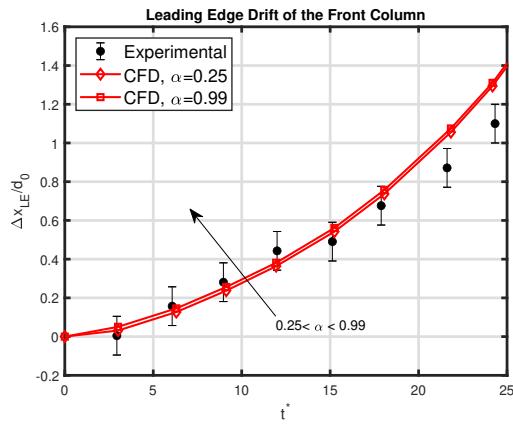


Figure 5.24: Normalized drift of the front column.

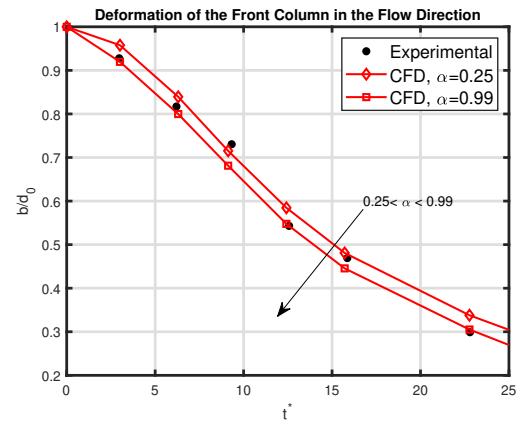


Figure 5.25: Normalized deformation of the front column.

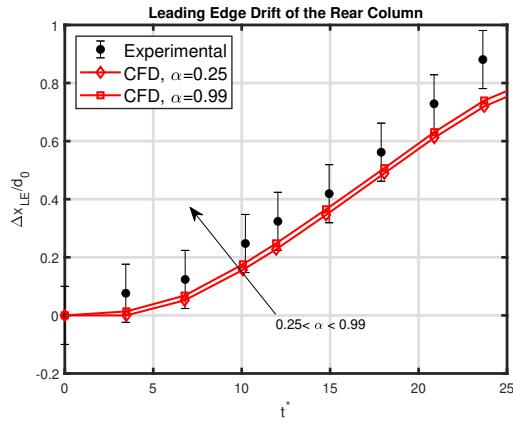


Figure 5.26: Normalized drift of the rear column.

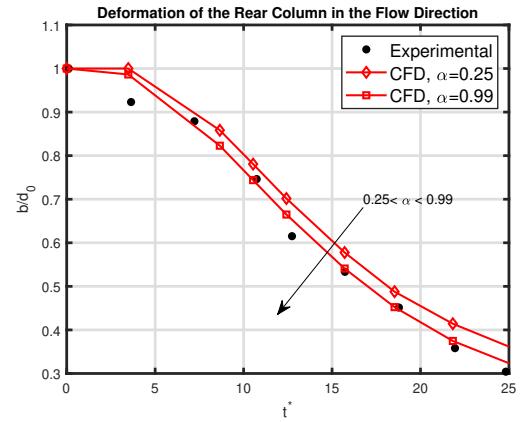


Figure 5.27: Normalized deformation of the rear column.

The displacement of the water columns in the tandem configuration was measured at the frontal stagnation point of each water column. The displacement of the front water column is similar to that of a single water column 5.12, whereas the displacement of the rear water column appears to be below that of the front one, as in the reference study. These results are also similar to what has been obtained on spherical drops [32]. However, in the late stages ($t^* = 46$) of the interaction, spherical droplets merge whereas water columns take more time. This is due to the frictional force that allow the water to adhere to the glass window in the experiment. This additional force, is also to be addressed where comparing the numerical and experimental results.

According to Igra and Takayama, calculated trajectories can be fitted using the equation

$$x^* = \frac{1}{2} a_t^{*2} \quad (5.5)$$

Moreover, with the following two, really poor assumptions, made while evaluating the drag coefficient of a spherical droplet:

- the water column shape does not change and the mass is conserved
- the acceleration of the water column is constant

one can obtain a first order approximation of the drag coefficient. The coupling of the assumptions with the equation of motion gives

$$\frac{1}{2} C_D \rho_2 u^2 S = ma$$

and, with the acceleration being

$$a = \frac{2}{\pi} \frac{C_d}{d_0} \frac{\rho_2}{\rho_1} u^2 \quad (5.6)$$

the dimensionless form of the relation is the following ($T^* = t^* \cdot \sqrt{\rho_2 / \rho_1}$):

$$x^* = \frac{1}{\pi} C_D T^{*2} \quad (5.7)$$

And the unsteady drag coefficient calculated from the x^* values are shown in Figure 5.28.

Although some values are overestimated, the integral average of the two fitted curves shown in figure 5.28, calculated as

$$\overline{C_D} = \frac{1}{\Delta T^*} \int_{T_0^*}^{T_0^* + \Delta T^*} C_D(T^*) d\tau \quad (5.8)$$

where the integral is approximated with the trapezoidal rule, are respectively 3.71 for the front column and 1.90 for the rear column. The theoretical values given by Igra and Takayama are ~ 3 and ~ 1.93 . Figure 5.29 shows the fitting of the trajectories using the theoretical C_D compared to the numerical results.

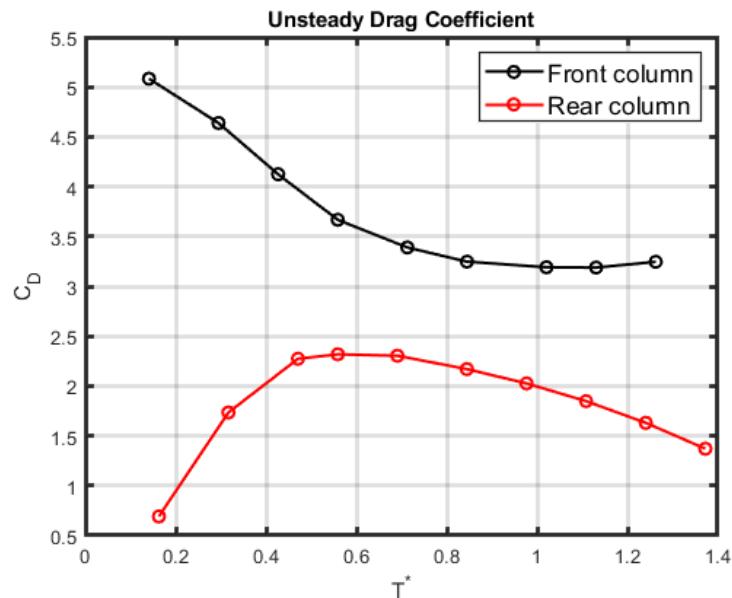
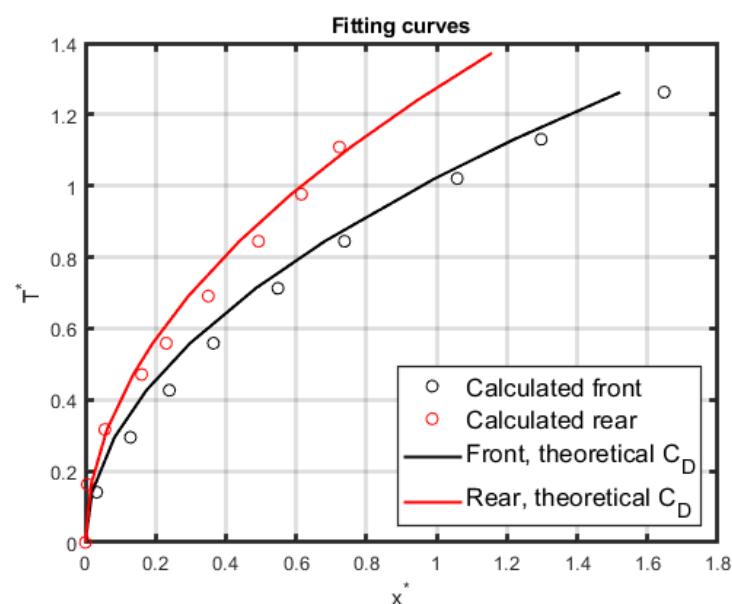
Figure 5.28: Unsteady C_D .

Figure 5.29: Fitting curves.

In addition, qualitative results were compared. Velocity/vorticity magnitude and pressure/Schlieren contours at significant instants are shown below (Figures 5.30, 5.31).

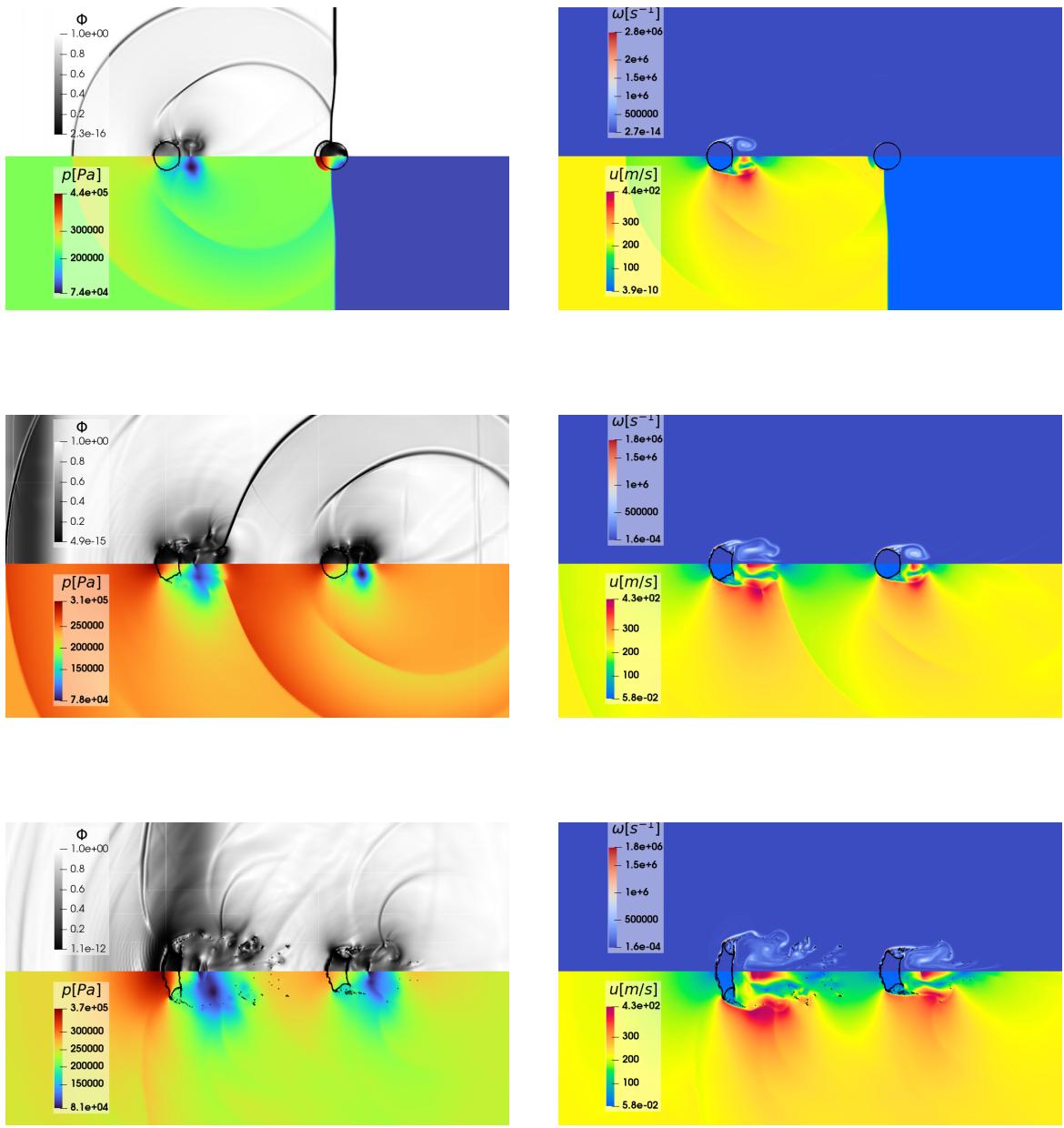


Figure 5.30: $t_1^* = 3.137$, $t_2^* = 6.335$, $t_3^* = 12.578$.

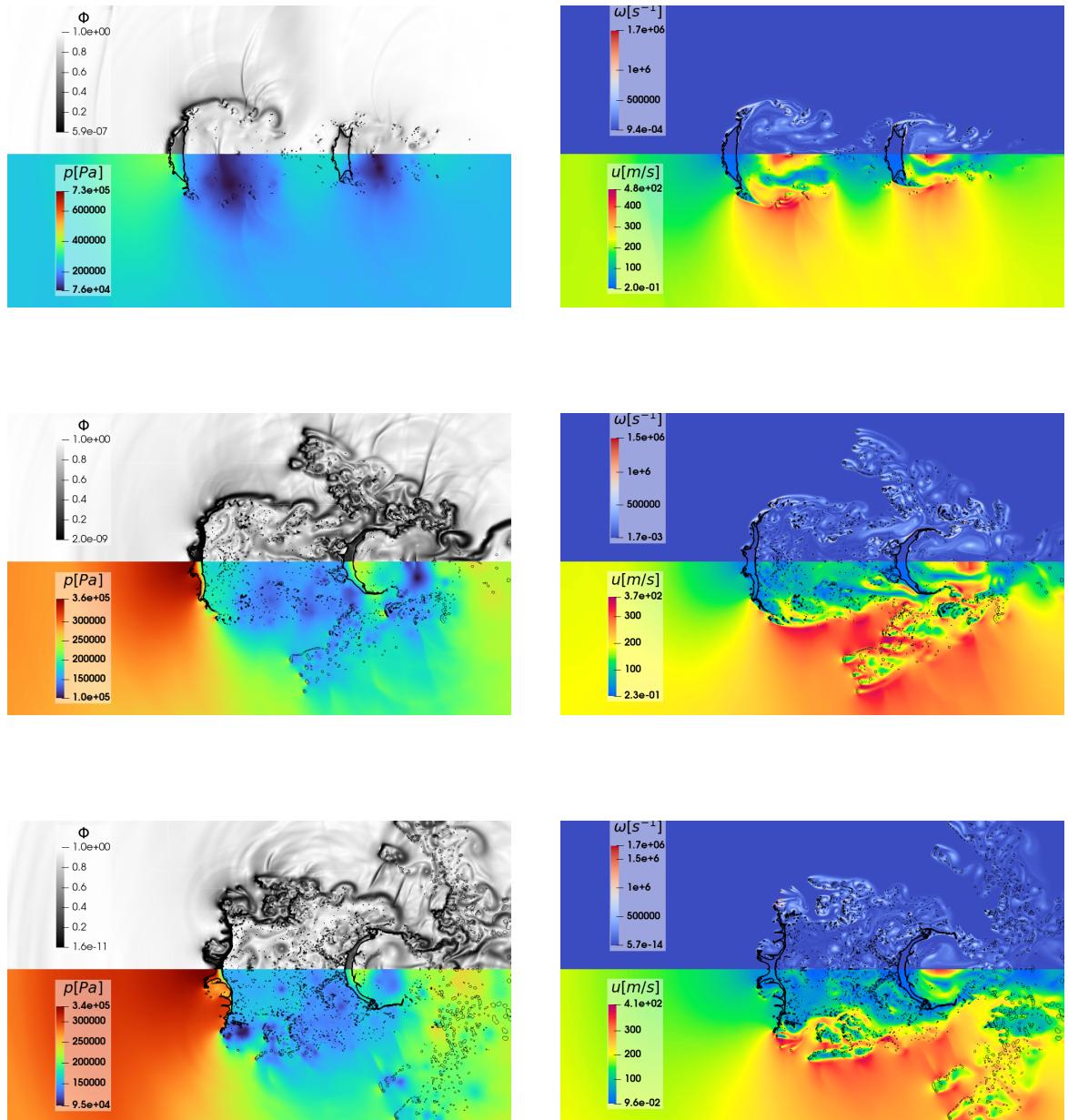


Figure 5.31: $t_1^* = 15.919$, $t_2^* = 25.544$, $t_3^* = 30.885$.

5.3.1 Kelvin-Helmholtz instabilities and Shear-Induced Entrainment

The initial stages of the interaction involving wave dynamics have been previously discussed. This subsection focuses on understanding the breakup dynamics and highlighting the key features of the SIE mode [31].

The initial stages of breakup are similar for all cases, and the cylinder attains a *cupcake shape*. After that, the droplet undergoes a stripping-like breakup. The stripping of daughter droplets near the equator of the primary droplet is triggered by the formation of KH waves near the front stagnation point. Those instabilities arise specifically at the interfaces between two fluids, where a *velocity shear*⁶ can be observed. The surface instabilities lead to a build-up of water near the equator: the internal flow of the droplet is directed towards the peripheral equatorial areas. A theoretical analysis shows that the subsequent daughter droplet formation increases with the increase in Weber number. Also, the size of daughter droplets increases during the entire SIE breakup mode. Theoretical studies to precisely characterize KH waves have also been carried, and can be found in [3, 31].

The vortices formation, leading to the peripheral stripping phenomenon, can be examined by plotting the vorticity, calculated using another instance of the python calculator to obtain the array of the velocity curl $\nabla \times \mathbf{u}$ (as it is been done for figure 5.21), and displaying its magnitude:

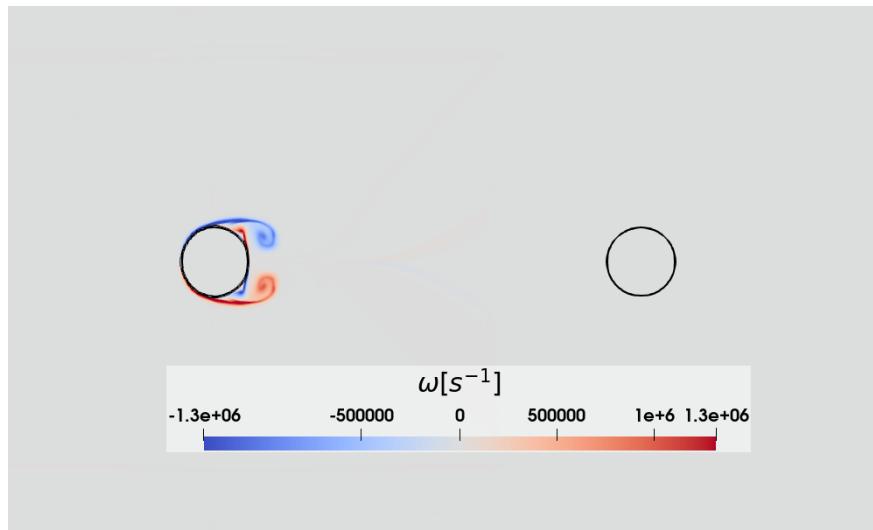


Figure 5.32: Vorticity on the $z = 0$ plane, $t^* = 2.10$.

⁶ Shear velocity is strictly related to shear stress

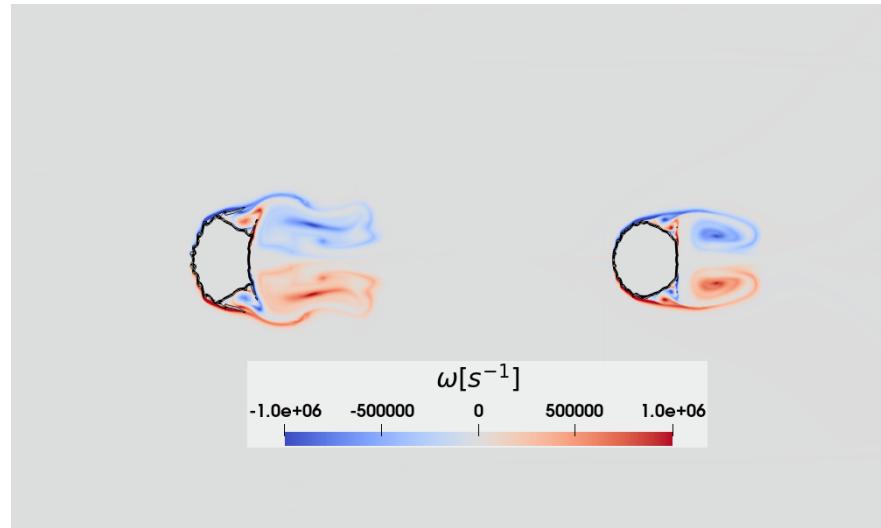


Figure 5.33: Vorticity on the $z = 0$ plane, $t^* = 6.9$.

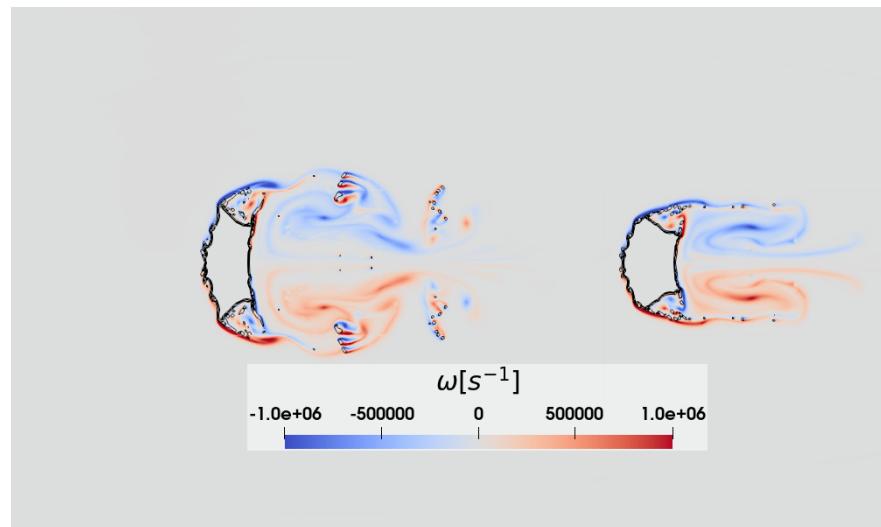


Figure 5.34: Vorticity on the $z = 0$ plane, $t^* = 11.05$.

This recirculation vortex appears to contribute to the stripping mechanism at the edge. As the cylinders flatten, the circulation zone in the wake region disappears. This phenomenon is facilitated by the wave reflected on the second cylinder.

5.4 NOTE: THERMODYNAMIC CONDITIONS OF WATER

Since the moving shock causes a high speed, high temperature, and high pressure flow to hit the droplet, it is good practice to check whether the water is still liquid⁷. Given $M = 1.47$

$$p_2 = 2.35 \cdot 101.3 \times 10^3 \text{ Pa} = 238.06 \times 10^3 \text{ Pa} \quad (5.9)$$

$$T_2 = 381.10 \text{ K} = 107.95^\circ \text{C} \quad (5.10)$$

It is possible to check the value of the vapor pressure with the *Antoine* equation:

$$p^\circ = 6.11 \cdot 10^{\frac{7.5 \cdot T_2}{237.7 + T_2}} = 6477 \text{ mbar} = 6.5 \text{ bar} = 6.5 \times 10^2 \text{ Pa} \quad (5.11)$$

Now for $M = 1.73$:

$$p_2 = 3.32 \cdot 101.3 \times 10^3 \text{ Pa} = 336.32 \times 10^3 \text{ Pa} \quad (5.12)$$

$$T_2 = 433.86 \text{ K} = 160.71^\circ \text{C} \quad (5.13)$$

$$p^\circ = 6.11 \cdot 10^{\frac{7.5 \cdot T_2}{237.7 + T_2}} = 1323 \text{ mbar} = 1.3 \text{ bar} = 1.3 \times 10^2 \text{ Pa} \quad (5.14)$$

Since $p^\circ < p_2$ in both cases there is no phase change. Otherwise, a model for cavitation would have been necessary.

⁷ This condition has been checked before setting the numerical model and it is only presented later as a note.

5.5 CONCLUSIONS

In conclusion, an alternative approach to the classical commercial softwares and stock OpenFOAM solvers has been proposed. Recalling what was the main focus of this work from the principle, the hybrid method has been validated and can effectively be used to simulate multiphase compressible interactions. The streamwise deformation and drift of the leading edge is predicted with sufficient accuracy. Moreover, the center of mass drift and velocity are very close to the in-house reference results. However, an important point has to be stated: as the non-dimensional analysis suggests, the essential physical characteristics that dominate the interaction and the deformation change. Thus, the numerical method used for the simulation has to change accordingly. Being very problem-dependent, this requires further extensive research.

As of the future perspectives, several key areas need attention, starting from the inherent weaknesses of the solver:

- different thermodynamic models, such as the *stiffenedGas* equation of state could be useful;
- current handling of the surface tension is similar to *interFoam*; nevertheless, dealing with that force without instabilities in the zones where the effect of the compressibility cannot be ignored is still a challenge;

For the late-stage interaction, employing a suitable turbulence model after thorough validation will be essential. While this thesis has focused on simulating the entire two-dimensional interaction, future improvements will stem from extending these simulations to fully three-dimensional interactions without relying on symmetry assumptions.

Regarding the simulation workflow, some ideas can be investigated to improve the efficiency and make better use of the computational resources:

- adaptive mesh refinement is being developed and could significantly improve the efficiency of the simulations;
- the correct use of NRBCs can allow for significant reduction of the computational domain

BIBLIOGRAPHY

- [1] M. Kravoshin *et al.*, ‘An extension of the all-mach number pressure-based solution framework for numerical modelling of two-phase flows with interface’, *Industrial processes and technologies*, vol. 2, 2022. doi: [10.37816/2713-0789-2022-2-3\(5\)-6-27](https://doi.org/10.37816/2713-0789-2022-2-3(5)-6-27).
- [2] J. H. Ferziger and M. Peric, *Computational Methods for Fluid Dynamics*. 2002.
- [3] E. Villermaux, ‘Fragmentation’, *Annual Review of Fluid Mechanics*, vol. 39, 2007. doi: <https://doi.org/10.1146/annurev.fluid.39.050905.110214>.
- [4] E. E. Timm and F. G. Hammitt, ‘A repeating water gun device for studying erosion by water jet impacts’, University of Michigan, 1969.
- [5] B. Moylan *et al.*, ‘Investigation of the physical phenomena associated with rain impacts on supersonic and hypersonic flight vehicles’, *Procedia Engineering*, vol. 58, 2013.
- [6] M. Mahdipoor *et al.*, ‘Influence of impact speed on water droplet erosion of TiAl compared with Ti6Al4V’, *Sci Rep*, 2015. doi: <https://doi.org/10.1038/srep14182>.
- [7] T. G. Theofanous and G. J. Li, ‘On the physics of aerobreakup’, *Physics of Fluids*, vol. 20, 2008.
- [8] W. G. Reinecke *et al.*, ‘Raindrop breakup in the shock layer of a high-speed vehicle’, *AIAA Journal*, vol. 10, no. 9, 1972. doi: [10.2514/3-50350](https://doi.org/10.2514/3-50350).
- [9] D. D. Joseph *et al.*, ‘Breakup of a liquid drop suddenly exposed to a high-speed airstream’, *International Journal of Multiphase Flow*, vol. 25, no. 6, 1999. doi: [https://doi.org/10.1016/S0301-9322\(99\)00043-9](https://doi.org/10.1016/S0301-9322(99)00043-9).
- [10] J. A. Nicholls and A. A. Ranger, ‘Aerodynamic shattering of liquid drops.’, *AIAA Journal*, vol. 7, no. 2, 1969. doi: [10.2514/3-5087](https://doi.org/10.2514/3-5087).
- [11] O. G. Engel, ‘Fragmentation of waterdrops in the zone behind an air shock’, *Journal of Research of the National Bureau of Standards*, vol. 60, no. 3, 1958.
- [12] A. Kapila *et al.*, ‘Two-phase modeling of deflagration-to-detonation transition in granular materials: Reduced equations’, *Phys. Fluids*, vol. 13, pp. 3002–3024, 2001. doi: [10.1063/1.1398042](https://doi.org/10.1063/1.1398042).
- [13] G. Allaire *et al.*, ‘A five-equation model for the simulation of interfaces between compressible fluids’, *Journal of Computational Physics*, vol. 181, no. 2, 2002. doi: <https://doi.org/10.1006/jcph.2002.7143>.
- [14] J. Meng and T. Colonius, ‘Numerical simulations of the early stages of high-speed droplet breakup’, *Shock Waves*, vol. 25, 2014. doi: [10.1007/s00193-014-0546-z](https://doi.org/10.1007/s00193-014-0546-z).
- [15] F. H. Harlow and A. A. Amsden, *Fluid Dynamics. A Lasl Monograph*. 1971.
- [16] S. P. Marsh, *LASL Shock Hugoniot Data*. University of California Press, 1980.
- [17] J. C. Meng and T. Colonius, ‘Numerical simulation of the aerobreakup of a water droplet’, *Journal of Fluid Mechanics*, vol. 835, 2018. doi: [10.1017/jfm.2017.804](https://doi.org/10.1017/jfm.2017.804).
- [18] F. Denner and B. Wachem, ‘Compressive vof method with skewness correction to capture sharp interfaces on arbitrary meshes’, *Journal of Computational Physics*, vol. 279, 2014. doi: [10.1016/j.jcp.2014.09.002](https://doi.org/10.1016/j.jcp.2014.09.002).
- [19] C. W. Hirt and B. D. Nichols, ‘Volume of fluid (vof) method for the dynamics of free boundaries’, *Journal of Computational Physics*, vol. 39, no. 1, 1981. doi: [https://doi.org/10.1016/0021-9991\(81\)90145-5](https://doi.org/10.1016/0021-9991(81)90145-5).
- [20] S. V. Poplavski *et al.*, ‘On the interaction of water droplet with a shock wave: Experiment and numerical simulation’, *International Journal of Multiphase Flow*, vol. 127, 2020. doi: <https://doi.org/10.1016/j.ijmultiphaseflow.2020.103273>.
- [21] F. Denner *et al.*, ‘Pressure-based algorithm for compressible interfacial flows with acoustically-conservative interface discretisation’, *Journal of Computational Physics*, vol. 367, 2018. doi: <https://doi.org/10.1016/j.jcp.2018.04.028>.
- [22] E. F. Toro *et al.*, ‘Restoration of the contact surface in the hll-riemann solver’, *Shock Waves*, vol. 4, 1994. doi: <https://doi.org/10.1007/BF01414629>.
- [23] M. Kravoshin *et al.*, ‘Adaptation of kurganov-tadmor numerical scheme for applying in combination with the piso method in numerical simulation of flows in a wide range of mach numbers’, *Procedia Computer Science*, vol. 66, 2015. doi: [10.1016/j.procs.2015.11.007](https://doi.org/10.1016/j.procs.2015.11.007).

- [24] C. Greenshields *et al.*, ‘Implementation of semi-discrete, non-staggered central schemes in a colocated, polyhedral, finite volume framework, for high-speed viscous flows’, *International Journal for Numerical Methods in Fluids*, vol. 63, 2009. doi: [10.1002/fld.2069](https://doi.org/10.1002/fld.2069).
- [25] C. Traxinger, ‘Real-gas effects and single-phase instabilities during injection, mixing and combustion under high-pressure conditions’, Ph.D. dissertation, Universität der Bundeswehr München, 2021.
- [26] P. Bohorquez, ‘Computational continuum mechanics for sediment transport in free-surface flow’, 2008.
- [27] L. Lucchese, ‘Implementation of non-reflecting boundary conditions in openfoam’, doi: http://dx.doi.org/10.17196/OS_CFD#YEAR_2022.
- [28] B. J. Cantwell, *Fundamentals of compressible flow*. 2022.
- [29] G. Sod, ‘A survey of several finite difference methods for systems of nonlinear hyperbolic conservation laws’, *J. Comput. Phys.*, vol. 27, pp. 1–31, 1978. doi: [10.1016/0021-9991\(78\)90023-2](https://doi.org/10.1016/0021-9991(78)90023-2).
- [30] K. Takayama and D. Igra, ‘Investigation of aerodynamic breakup of a cylindrical water droplet’, *Atomization and Sprays*, 2001. doi: [10.1615/AtomizSpr.v11.i2.50](https://doi.org/10.1615/AtomizSpr.v11.i2.50).
- [31] S. Sharma *et al.*, ‘Shock induced aerobreakup of a droplet’, *Journal of Fluid Mechanics*, 2021. doi: [10.1017/jfm.2021.860](https://doi.org/10.1017/jfm.2021.860).
- [32] D. Igra and K. Takayama, ‘Study of a cylindrical water column subjected to planar shock wave loading’, 2000. doi: [10.1115/IMECE2000-2044](https://doi.org/10.1115/IMECE2000-2044).