POLITECNICO DI MILANO

Scuola di Ingegneria Industriale e dell'Informazione

Dipartimento di Energia

Corso di Laurea Magistrale in Ingegneria Energetica

# Feasibility analysis of a two-fluid solver for cavitation and interface capturing as implemented in OpenFOAM

Relatore: Prof. Federico Piscaglia

Tesi di Laurea Magistrale di:
Zeno Tacconi Matr. 849956

**Anno Accademico 2017-2018**

Politecnico di Milano:
www.polimi.it
Scuola di Ingegneria Industriale e dell'Informazione:
www.ingindinf.polimi.it

*"The boundaries which divide Life from Death*
*are at best shadowy and vague.*
*Who shall say where the one ends, and where the other begins"*
***Edgar Allan Poe, The Premature Burial***

# Ringraziamenti

Desidero innanzitutto ringraziare il Prof. Federico Piscaglia per la gentilezza e la disponibilità dimostrata nei miei confronti durante tutto il periodo di tesi. Lo ringrazio per i pomeriggi e le serate spese a trasmettermi le basi per comprendere la CFD e C++ e soprattuto per aver condiviso con me la soddisfazione di ogni successo ma anche la frustrazione di ogni insuccesso.

Un grazie speciale va anche allo studente di dottorato Filippo Giussani che mi ha accolto nel dipartimento di energia facendomi sentire come a casa e con la sua generosità ha sempre trovato del tempo per aiutarmi quando "nulla convergeva". Lo ringrazio in particolare per avermi insegnato come "non programmare" in C++.

Ringrazio i miei genitori, mio fratello, la mia ragazza e tutta la mia famiglia che mi hanno sempre sostenuto in questi anni di Politecnico (non solo economicamente), che hanno condiviso con me i momenti migliori e anche quelli peggiori di questa carriera universitaria e che mi hanno fatto ritornare nella mia cameretta ogni weekend, come nulla fosse cambiato.

Ringrazio tutti i "butei" da Verona a Milano fino all'Iran che hanno reso questi anni di Politecnico gli "anni più belli della mia vita" e che ricorderò sempre con affetto, dalle giornate di studio intenso alle serate pazze del Club Modena.

Ringrazio tutto il team del PMF (Polimi Motorcycle Factory) ed in particolare i ragazzi di simulazione 3D che hanno saputo accogliermi nel loro gruppo e con cui ho passato dei bellissimi pomeriggi di "elucubrazioni" in sala corsi.

Infine un ringraziamento a tutti i ragazzi della sala visiting che hanno aiutato a sdrammatizzare i momenti più difficili, perchè si sa: "mal comune mezzo gaudio".

Zeno

# Acknowledgements

I would like to begin by thanking Prof. Federico Piscaglia for the kindness and willingness shown towards me during this thesis. I thank him for the afternoons and evenings spent teaching me the basis of CFD and C++ and especially for sharing with me the satisfaction of any success as well as the frustration of any failure.

Particular thanks also goes to the PhD student Filippo Giussani, who has welcomed me in the Department of Energy making me feel like home and with his generosity was able to find always some time to help me when "nothing converged". I thank him in particular for teaching me how "not to program" in C++.

I would like to thank my parents, my brother, my girlfriend an all my family who have always supported me during these years (not just economically), who have shared with me the best and the worst moments during my academic career and who brought me back to my old bedroom almost every weekend, like nothing had changed.

I thank the "butei" from Verona to Milan and even Iran who have mad these years the best of my life. I will always remember very fondly, from the heavy-study days to the crazy nights spent at Club Modena.

I thank the PMF (Polimi Motorcycle Factory) team and in particular the guys of 3D simulation who were able to welcome me in their group and with whom I spent long afternoons of "lucubrations" in sala corsi.

Finally, I thank my fellow labmates from sala visiting who helped me minimize the tough times, because it is known that:"Two in distress make sorrow less".

Zeno

# Extended Abstract

## Introduction

Multiphase flows are adopted in several industrial applications such as power plants [1, 2], oil and gas field management [3], combustion and in general chemical processes [4], cavitation in technical applications [5, 6]. The study of multiphase flows and in particular of two-phase fluids, which will represent the main subject of this work, shows a large amount of variants that are worth being discussed. The direct measurement of characteristic flow variables such as velocity, volume fraction and temperature are not always possible mainly because the process could be altered by the acquisition method or because the equipment to be tested is too small to fit any experimental system of measurement, as in the case of the study of internal nozzle flows in fuel injectors. Obtaining experimental evidences from this kind of processes is really cumbersome and most of the time includes the use of pilot studies (experiments carried out in thermo-hydrodynamic similitude with the case study) [7–9] as well as empirical correlations. Nowadays, CFD simulations of internal nozzle flows are becoming very popular and are performed in combination to experimental studies, to expand the applicability of the solution method to a wider spectrum. Among the several CFD methodologies available in the literature, the VoF method (Volume of Fluid) is preferred for simulations of injectors [7, 9–12]. Main drawback of this methodology is the lack of information about the two fluids since a single momentum equation is solved. In 2016 Edelbauer [10] proposed the validation of an hybrid solver: a three-phase system consisting of liquid, vapour and gas is applied for the Volume of Fluid simulation of the liquid disintegration in order to resolve the liquid-gas interface. The method presented by Edelbauer overcomes the problem of high computational effort needed by surface-capturing methods (each bubble is resolved singularly) by modelling the liquid-vapour interface within the Eulerian Eulerian framework, while the liquid-gas interface is described by the VoF method. With this approach Edelbauer was able to reach (with reasonable computational effort) a more accurate solution with respect

to Sou's one. Some results are plotted in figure 1 and show that the vapour fraction, even at small pressure jump, is intercepted more accurately.

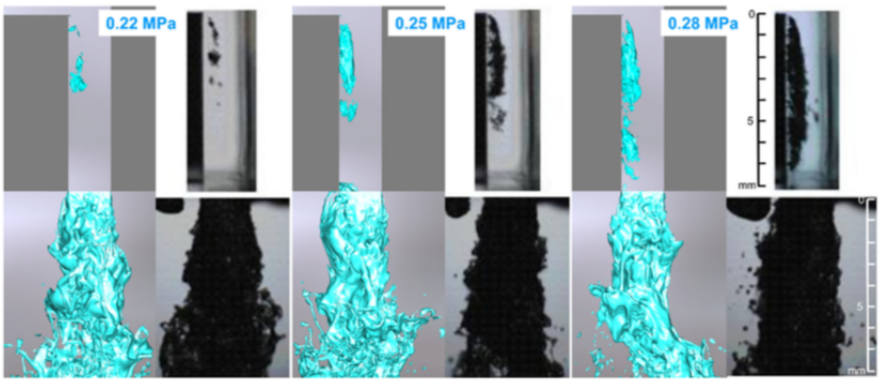


Figure 1: Comparison of simulated liquid volume fraction iso-surface snapshots ($\alpha_{iso} = 0.5$) and instantaneous photographs of the experiment from [7] for $p_{in} = 0.22$, 0.25 and $0.28 MPa$.

The work of [10] evidences two fundamental characteristics that are preferable for a multiphase solver, namely the ability to capture the gas/liquid interface and to account for the slip velocity at the gas-liquid interface.

The aim of this work is then to evaluate the feasibility of the application of an Eulerian-Eulerian solver to simulate cavitation in injector geometries. The solver `reactingTwoPhaseEulerFoam` in the OpenFOAM software was selected for the study; the solver, originally designed for multiphase chemical reactors and fluidized-bed combustors, is able to track the mass fraction of the chemical species and to account for chemical reactions. With this in mind, the interconnection between the species transport equations and the Eulerian-Eulerian framework transport equations has been investigated, with particular focus on the correlation between the mass source term and the transport of the volume (void) fraction between different phases. Finally, the metodology employed in OpenFOAM for interface tracking in multiphase solvers was studied and validated on a numerical experiment; a quantitative evaluation and a complete analysis on benchmark quantities defined in [13] was performed; comparison with a two-fluid VoF solver `interFoam` are provided.

# Governing Equations

In order to better understand how the two fluid model describes the two phases it is following presented the formulation of the transport equation

defining the model. The formulation of the phase transport equation is based on the conditional averaging of the continuity equation based on the work of Weller [14] as described in [15].

$$\frac{\partial \rho_\varphi \alpha_\varphi}{\partial t} + \nabla \cdot (\rho_\varphi \vec{\bar{U}}_\varphi \alpha_\varphi) = \Gamma_\varphi \tag{1}$$

where $\varphi$ identifies the phase, $\alpha$ is the phase fraction and $\Gamma$ represents the mass source term. It is possible to write a phase-continuity equation for each of the two phases ($\varphi \in [1, 2]$) and combine them to obtain the volumetric continuity equation for the mixture. Weller [14] proposed a different formulation of the phase continuity equation, so that all terms are conservative and $\alpha_1$ is bounded, introducing two different velocities $U$ and $U_r$.

$$\vec{\bar{U}} = \alpha_1 \vec{\bar{U}}_1 + \alpha_2 \vec{\bar{U}}_2 \tag{2}$$

$$\vec{\bar{U}}_r = \vec{\bar{U}}_1 - \vec{\bar{U}}_2 \tag{3}$$

This rearranged equation couples implicitly the two phases through the introduction of these two velocities, applying Weller's formulation to both phases and rearranging the equations, we obtain:

$$\begin{aligned} \frac{\partial(\alpha_1)}{\partial t} + \nabla \cdot (\alpha_1 \vec{\bar{U}}) + \nabla \cdot \left(\alpha_1 \vec{\bar{U}}_r (1 - \alpha_1)\right) - \alpha_1 \nabla \cdot (\vec{\bar{U}}) = \\ \alpha_1 \alpha_2 \left( \frac{1}{\rho_2} \frac{D(\rho_2)}{Dt} - \frac{1}{\rho_1} \frac{D(\rho_1)}{Dt} \right) \end{aligned} \tag{4}$$

Following the analysis carried on the continuity equation, conditional averaging is performed also on the momentum equation derived from the generalised transport equation for two-phase flows:

$$\frac{\partial(\alpha_\varphi \rho_\varphi \vec{\bar{U}}_\varphi)}{\partial t} + \nabla \cdot (\alpha_\varphi \rho_\varphi \vec{\bar{U}}_\varphi \vec{\bar{U}}_\varphi) + \nabla \cdot (\alpha_\varphi \bar{\mathcal{R}}_\varphi^{eff}) = -\alpha_\varphi \nabla \bar{p} + \alpha_\varphi \rho_\varphi \vec{g} + \bar{\mathcal{M}}_\varphi \tag{5}$$

where $\bar{\mathcal{R}}_\varphi^{eff}$ is the combined Reynolds and viscous stress tensor [16, 17] and $\bar{\mathcal{M}}_\varphi$ represents the averaged inter-phase momentum transfer term for both dispersed and segregated flows. This last term has to be defined, and hence modelled for each phase. In order to obtain a conservative solution the total momentum transferred through phases has to be null. This means that the summation of each momentum transfer contribution for both phases has to be zero:

$$\sum_{\substack{\varphi \in [1,2] \\ i \in [IMTM]}} \bar{\mathcal{M}}_{\varphi_i} \tag{6}$$

In a two-fluid approach, it is sufficient to define one expression for the inter-phase momentum transfer to reach closure of the system. Usually the models are fitted to describe the forces acting on the dispersed phase, or better to the Dispersed Phase Elements.

# Interfacial Momentum Transfer Modelling

In order to correctly assess the effect of interfacial momentum transfer it is also worth to highlight the substantial mathematical and physical differences between segregated and dispersed flows. In this section the two diverse framework will be described and the deriving inter-phase change terms will be listed. Starting from the momentum averaged transport equation (5) the source term will be described for both dispersed and segregated flows as the summation of two components namely the net force contribution from interfacial averaged pressure in case there is a gradient in the volumetric phase fraction and the contribution due to shear stress and unbalanced interfacial pressure.

$$\bar{\mathcal{M}}_{\varphi} = \bar{\mathcal{M}}_{\sigma,\varphi} + \bar{\mathcal{M}}_{h,\varphi} \tag{7}$$

The first source term takes into account the phase slip and the surface tension effects, while the second term =0.25em

- for the dispersed two-fluid system represents the sum of fluid dynamic forces applied to a Disperse Phase Element (DPE):

$$\begin{aligned} \bar{\mathcal{M}}_1 &= A_d \vec{\bar{U}}_r && \text{Drag} \\ &+ A_{vm}\left(\frac{D_2\vec{\bar{U}}_2}{Dt} - \frac{D_1\vec{\bar{U}}_1}{Dt}\right) && \text{Virtual Mass} \\ &+ A_{l_2}\vec{\bar{U}}_r \times (\nabla \times \vec{\bar{U}}_2) && \text{Lift} \\ &+ A_{l_1}\vec{\bar{U}}_r \times (\nabla \times \vec{\bar{U}}_1) && \\ &+ A_{\alpha}\nabla\alpha && \text{Turbulent Dispersion} \\ &+ \mathcal{M}_1^{wl} && \text{Wall Lubrication} \end{aligned} \tag{8}$$

- for segregated flow types, it models the interfacial force density due to unbalanced pressures and viscous stresses, that lead to a dissipative drag due to interfacial friction in the presence of phase slip.

  In the case of Segregated flows the interfacial morphology is partially resolved, i.e. the averaging volume is chosen sufficiently in its size to capture the main (mean) dynamics of the interfacial flow, whereas non-resolved interfacial morphologies and phase interactions again must be accounted

for in the underlying averaging framework by appropriate (physical meaningful) closure models. As a consequence, the interface is represented as an interfacial transition region of certain characteristic width

$$\begin{aligned} \bar{\mathcal{M}}_{\varphi,\sigma} &= \alpha_\varphi \cdot 4\alpha_\varphi\alpha_\phi\bar{\mathcal{M}}_\sigma \\ \bar{\mathcal{M}}_{\phi,\sigma} &= \alpha_\phi \cdot 4\alpha_\phi\alpha_\varphi\bar{\mathcal{M}}_\sigma \end{aligned} \tag{9}$$

with:

$$\begin{aligned} \bar{\mathcal{M}}_\sigma &= \sigma\widetilde{\mathcal{K}_I}\widetilde{n_I}\Sigma + \sigma\frac{d\Sigma}{d\alpha_\varphi}\nabla\alpha_\varphi \\ &\approx \left(\nabla\cdot\left(\frac{\nabla\alpha_\varphi}{|\nabla\alpha_\varphi|}\right) + 4\Sigma_0(1-2\alpha_\varphi)\right)\nabla\alpha_\varphi \end{aligned} \tag{10}$$

For what concerns the interfacial drag term it is modelled as follows:

$$\bar{\mathcal{M}}_{h,\varphi} = \lambda(Re_I,\pi_\mu)\frac{|\nabla\alpha_\varphi|}{\delta}\frac{\bar{\mu}^\varphi\bar{\mu}^\phi}{\bar{\mu}^\varphi+\bar{\mu}^\phi}\left(\bar{U}^\varphi-\bar{U}^\phi\right) \tag{11}$$

Where $\bar{\mu}$ is the field average viscosity, $|\nabla\alpha_\varphi|$ is used to model $\Sigma$, i.e. the interfacial area density that varies across the interfacial transition region of width $\delta$. $\lambda(Re_I,\pi_\mu)$ is a proportionality factor representing the dimensionless friction coefficient which holds tangential inertia and tangential shear contributions:

$$\lambda(Re_I,\pi_\mu) = mRe_I + n\pi_\mu \tag{12}$$

the parameters $m$ and $n$ have to be appropriately chosen ($0.1 \leq m \leq 1.5$ and $n \approx 8$ ).

# Cases Setup

The initial configuration is identical for both cases and consists of a circular bubble of diameter $d_0 = 0.5$ centred at $[0.5; 0.5]$ in a $[1 \times 2]$ rectangular domain, as represented in figure 2. The density of the bubble is lower than that of the surrounding fluid ($\rho_b < \rho_l$). No-slip boundary conditions are applied on top and bottom walls ($\vec{u} = \vec{v} = 0$), while the free slip condition is applied to the vertical walls ($\vec{u} = 0$).

Table 1 lists the fluids and physical parameters which specify the test cases. The evolution of the bubbles should be tracked for 3 time units during which the defined benchmark quantities should be measured. Adimensionalization is frequently used to assist with classifying simulations and can be introduced by scaling through a characteristic length and time:

$$L = d_0$$

$$t = \frac{L}{U_g}$$

where $U_g = \sqrt{g d_0}$ is the gravitational velocity.

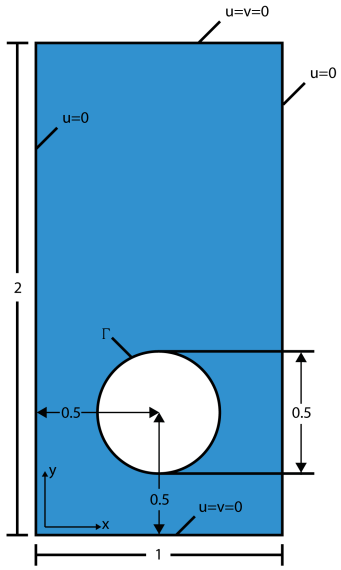


Figure 2: Initial configuration and boundary conditions for the test cases.

Table 1: Pysical parameters defining the test cases

| Test Case | $\rho_l$ | $\rho_g$ | $\mu_l$ | $\mu_g$ | g | $\sigma$ | $\mathcal{R}e$ | $\mathcal{E}o$ | $\rho_l/\rho_g$ | $\mu_l/\mu_g$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1000 | 100 | 10 | 1 | 0.98 | 24.5 | 35 | 10 | 10 | 10 |
| 2 | 1000 | 1 | 10 | 0.1 | 0.98 | 1.96 | 35 | 125 | 1000 | 100 |

It is worth to mention that the VoF solver of reference was `interFoam`, used as the benchmark solution for the comparison described in this paper, while the Euler-Euler solver of reference was `twoPhaseEulerFoam`, it was preferred since the benchmark cases are non reacting/cavitating.

## Case 2

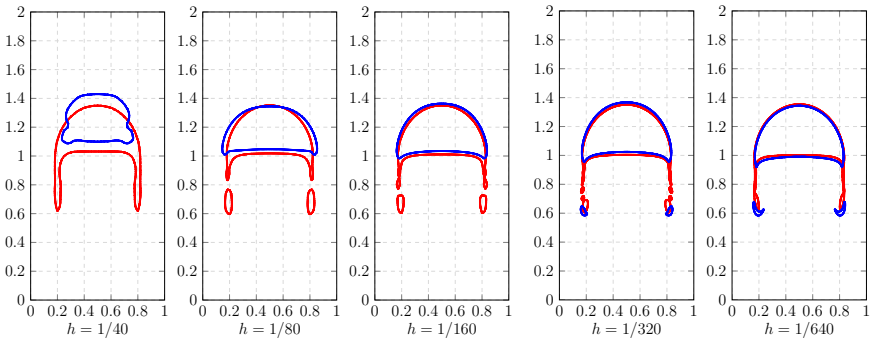


Figure 3: Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions.

The contour chart reported in figure 3 shows for simulations of Case 2 a lack of cohesion in coarsest meshes that manifest in an early breakup followed by a unphysical interface tracking. This phenomenon is visible as well from the bubble circularity chart, in which the concavity change corresponds to the bubble breakup. The unphysicality of coarsest mesh solutions with `twoPhaseEuerFoam` is confirmed by the trends shown in Fig. 5, where severe deviation from the benchmark trends can be spotted. It is also interesting to notice that with mesh refinement the rising velocity and the bubble centroid are simulated correctly. This suggests that even if the interface is not being tracked correctly the momentum equation solution is correct and provides the right velocity value. The bubble centroid is located almost at the "core-bubble" centroid, since that the deviations in the barycenter position due to the bubble-tails are light. Figure 4 shows how substantially convergence is reached in both solvers towards the finest mesh solution and provide a numerical comparison between the two solutions.

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.5594 | 0.53374 | 0.51518 | 0.49057 | 0.49252 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max,1}$ | 0.2411 | 0.24622 | 0.24876 | 0.2487 | 0.2506 |
| $t\|_{v=v_{max,1}}$ | 0.70937 | 0.72031 | 0.71875 | 0.72031 | 0.7210 |
| $v_{max,2}$ | 0.21598 | 0.22525 | 0.23125 | 0.23474 | 0.23694 |
| $t\|_{v=v_{max,2}}$ | 1.7406 | 1.8797 | 1.9688 | 2.0094 | 2.0105 |
| $y_c\|_{t=3}$ | 1.0945 | 1.1059 | 1.1161 | 1.1219 | 1.1291 |

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.65637 | 0.64369 | 0.6167 | 0.595 | 0.5702 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max,1}$ | 0.28816 | 0.26418 | 0.25739 | 0.25295 | 0.24972 |
| $t\|_{v=v_{max,1}}$ | 0.62 | 0.66 | 0.7 | 0.71 | 0.7 |
| $v_{max,2}$ | 026759 | 0.23716 | 0.22766 | 0.23672 | 0.24102 |
| $t\|_{v=v_{max,2}}$ | 1.04 | 1.64 | 1.85 | 2 | 2.1 |
| $y_c\|_{t=3}$ | 1.1131 | 1.102 | 1.1231 | 1.1317 | 1.1320 |

Figure 4: Comparison between Vof's (left) and Euler-Euler's (right) charachteristic benchmark values

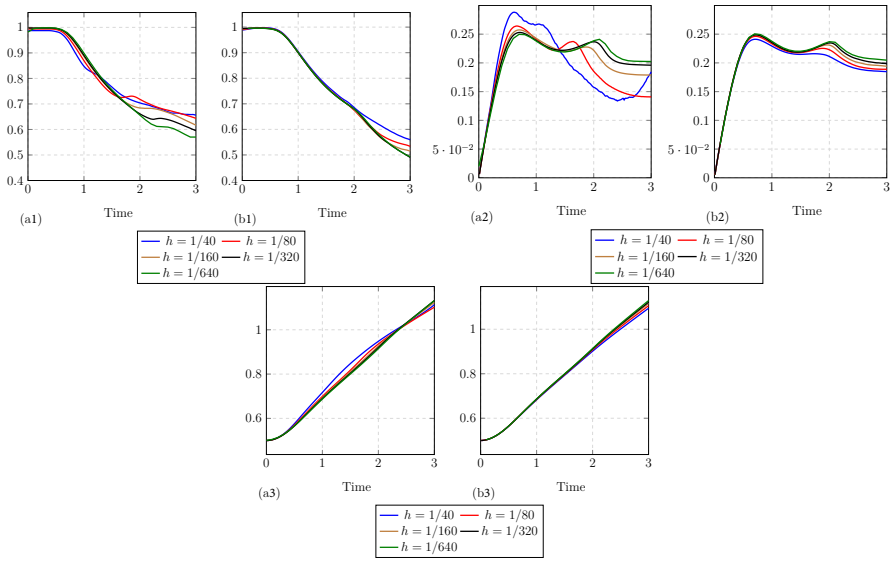


Figure 5: Comparison between bubble circularities at different mesh resolutions for both two-fluid method (a1) and VoF (b1). Comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a2) and VoF (b2). Comparison between bubble centroids at different mesh resolutions for both two-fluid method (a3) and VoF (b3).

## Case 1

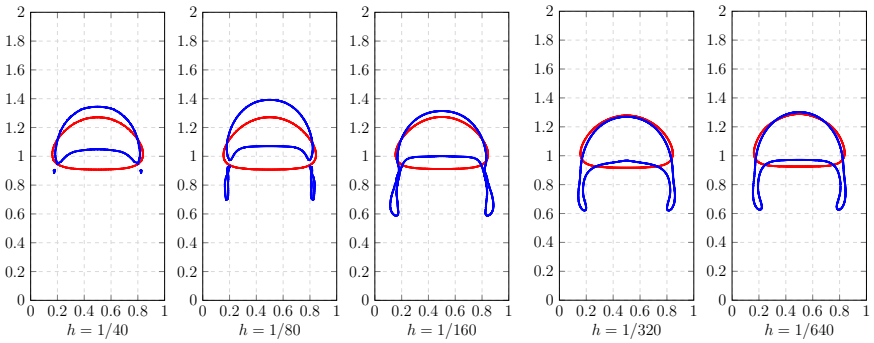


Figure 6: Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions.

From figure 6 it is evident that a unphysical solution is derived for all resolutions. Solutions of Case 1 simulations for `twoPhaseEulerFoam` show a lack of cohesion of the bubble, which shows a skirted profile with a set-up typical of

ellipsoidal range bubbles. The benchmark parameters are not comparable to the VoF solution except in some part for the finest meshes' rise velocity and bubble centroid position. This, as said before, is due to the correct solution of the momentum equations and the light deviation brought by the tails to the bubble centroid position. This simulation is vital for the analysis carried on in this paper, since it brings out that Euler-Euler method as implemented in `twoPhaseEulerFoam` is not suitable for interface-tracking simulations. Finally, figure 7 help demonstrate that convergence is not reached towards the finest solution in the Eulerian-Eulerian simulation, except in some part for the finest solution and provides a quantitative comparison between the two simulations.

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.5594 | 0.53374 | 0.51518 | 0.49057 | 0.49252 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max,1}$ | 0.2411 | 0.24622 | 0.24876 | 0.2487 | 0.2506 |
| $t\|_{v=v_{max,1}}$ | 0.70937 | 0.72031 | 0.71875 | 0.72031 | 0.7210 |
| $v_{max,2}$ | 0.21598 | 0.22525 | 0.23125 | 0.23474 | 0.23694 |
| $t\|_{v=v_{max,2}}$ | 1.7406 | 1.8797 | 1.9688 | 2.0094 | 2.0105 |
| $y_c\|_{t=3}$ | 1.0945 | 1.1059 | 1.1161 | 1.1219 | 1.1291 |

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.65637 | 0.64369 | 0.6167 | 0.595 | 0.5702 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max,1}$ | 0.28816 | 0.26418 | 0.25739 | 0.25295 | 0.24972 |
| $t\|_{v=v_{max,1}}$ | 0.62 | 0.66 | 0.7 | 0.71 | 0.7 |
| $v_{max,2}$ | 026759 | 0.23716 | 0.22766 | 0.23672 | 0.24102 |
| $t\|_{v=v_{max,2}}$ | 1.04 | 1.64 | 1.85 | 2 | 2.1 |
| $y_c\|_{t=3}$ | 1.1131 | 1.102 | 1.1231 | 1.1317 | 1.1320 |

Figure 7: Comparison between Vof's (left) and Euler-Euler's (right) charachteristic benchmark values

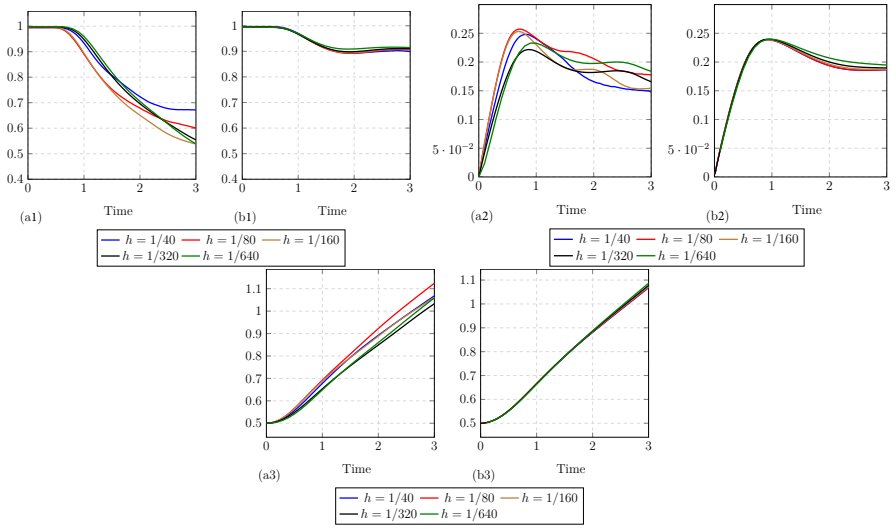


Figure 8: Comparison between bubble circularities at different mesh resolutions for both two-fluid method (a1) and VoF (b1). Comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a2) and VoF (b2). Comparison between bubble centroids at different mesh resolutions for both two-fluid method (a3) and VoF (b3).

# Conclusions

Recalling what said in the introduction about the requirements of a solver to be suitable for simulating cavitation in injectors and injector-like geometries, a two-fluid solver should capture gas/liquid interface correctly and be able to take into account the slip velocity between phases. Since cavitation is the most important source/sink phenomenon in this kind of problems, the solver should be capable to model also this contribution.

`reactingTwoPhaseEulerFoam` features a "*interfaceCompositionModel*" that allow to determine through a user-selected model the composition of the species at phase-interface. Hence the introduction of a homogeneous reactor model for cavitation in a two-fluid solver can be perform by taking advantage of the high modularity of the existing framework.

The results of this work brought to the author's attention that not only the two fluid Euler-Euler methodology is more mesh demanding than the VoF solver, used as benchmark solver , but also that severe diffusion and lack of cohesion manifest even at high mesh resolutions. This suggest that a modelling error is committed systematically by the solver, meaning that in *twoPhaseEulerFoam* the form of the terms for surface tension is not suitable for the case studied.
An accurate revision of the solver's code resulted in the finding that no equation for surface tension is currently implemented in OpenFOAM-dev. This somehow justifies the irrational results obtained from the simulations.

Several options can be implemented; the simplest barotropic model for cavitation, coupled to a homogeneous reactor model, should be implemented and the solver should be validated through a pilot study [7] and also compared to the Vof solver (`interPhaseChangeFoam`) solution. Once the cavitation phenomenon is demonstrated to be correctly tracked by the solver, a further development of the cavitation modelling could be studied. Together with this improvement in barotropic cavitation modelling other, more complex, models should be considered. In literature, as well as in OpenFOAM, models based on the Rayleigh-Plesset equation such as Schnerr and Sauer were demonstrated to be reliable to describe cavitation and widely validated [18].

Interface tracking in two fluid solver should be improved. It is suggested as a further development to implement the surface tension source term described in (10). The implementation should be consistent with the OpenFOAM formulation of two fluid method, in particular the source term should be activated only if the segregated version of the solver is being used.

# Abstract

The aim of this work is to evaluate the feasibility of the application of an Eulerian-Eulerian solver to simulate cavitation in injector-like geometries. In order to achieve this goal, the research started with the selection of the most suitable solver already implemented in OpenFOAM. Among the different two-fluid Eulerian-Eulerian solvers approach. Starting from the existing framework implemented in OpenFOAM, the aim was to implement a "homogeneous reactor model" for cavitation in a compressible two-fluid solver. With this in mind, most of the efforts were oriented to obtain a full description of the solver implementation and to outline differences with other solvers from the existing literature [2,15,19–21], when present. The link between the transport equations of the chemical species and the governing equations in the Eulerian-Eulerian framework was investigated, with particular focus on the correlation between the mass source term and the transport of the volume (void) fraction, to understand whether it is possible to correlate mass transport and volume fraction transport in a cavitation source term. The stencil of the equations reported in this thesis have been derived directly by the implementation of the equations in the software (OpenFOAM-dev, released by the OpenFOAM Foundation), to check the consistency with the published litterature, since no code documentation was available. Finally, the ability of the Euler-Euler solver to track the fluid interface is discused by a comparison with a Volume-Of-Fluid (VoF) solver, performed on a numerical experiment for validation available from the litterature.

**Keywords:** CFD, OpenFOAM, Euler-Euler, Interface Tracking, Cavitation,

`reactingTwoPhaseEulerFoam`

# Sommario

Questo lavoro punta a valuare la fattibilità di applicazione di un metodo Eulero-Eulero per la simulazione CFD di flussi bifase in iniettori. Per ottenere questo risultato, il lavoro di ricerca è cominciato con la selezione di un solutore già implementato in OpenFoam, tra i vari solutori Eulero-Eulero bifase. Partendo dal codice implementato in OpenFOAM l'obiettivo era di implementare un "modello di reattore omogeneo" per la cavitazione in un solutore comprimibile e bifase. Detto ciò, è importante sottolineare che la maggior parte del lavoro è stato orientato alla corretta descrizione del solutore come implementato e all'identificazione delle differenze del suddetto solutore con altri analoghi presenti in letteratura [2, 15, 19–21]. Particolare attenzione è stata posta nell'individuare il legame tra le equazioni di trasporto delle specie chimiche e le equazioni costitutive del modello Eulero-Eulero, più precisamente è stata ricercata una correlazione tra il termine sorgente dell'equazione del trasporto della massa e quello dell'equazione del trasporto della frazione in volume, per capire se fosse possibile correlare il trasporto della massa e il trasporto della frazione volumetrica con un termine sorgente di cavitazione. Le equazioni derivate in questa tesi sono state derivate direttamente dall'implementazione delle stess nel software (OpenFOAM-dev,rilasciato da OpenFOAM Foundation), per verificare la consistenza con la letteratura di riferimento, visto che nessua documentazione riguardo il codice è presente. Infine, le capacità di tracciare l'interfaccia tra due fluidi è discussa per il modello Eulero-Eulero attraverso un confronto con un solutore che usa l'approccio del "Volume dei Fluidi" (VoF). Questo confronto è stato attuato tramite un esperimento numerico per la validazione disponibile in letteratura.

**Parole Chiave:** CFD, OpenFOAM, Eulero-Eulero, Interfaccia, Cavitazione,

`reactingTwoPhaseEulerFoam`

# Contents


**1 Introduction** **1**
- 1 Background . . . 1
- 2 Multiphase-Flow Dynamics . . . 5
  - 2.1 Bubble Motion . . . 7
- 3 Objectives . . . 10
- 4 Thesis Outline . . . 11

**2 Two-Fluid Model** **13**
- 1 CFD Metodologies for Two-Phase Flow . . . 13
  - 1.1 Eulerian-Lagrangian Approach . . . 13
  - 1.2 Eulerian-Eulerian Approach . . . 15
- 2 Local Instantaneous Bulk Conservation Equations . . . 17
  - 2.1 Single-Phase Flow . . . 17
  - 2.2 Two-Phase Flow . . . 20
- 3 Governing Equations . . . 22
  - 3.1 Averaged Continuity Equation . . . 22
  - 3.2 Averaged Momentum Equation . . . 24
  - 3.3 Averaged Energy Equation . . . 24
  - 3.4 Turbulence Modeling . . . 26
- 4 Two-Phase Cavitating Systems . . . 28
  - 4.1 Continuity Equation . . . 28
  - 4.2 Momentum Equation . . . 28
  - 4.3 Energy Equation . . . 28
- 5 Interfacial Momentum Transfer Modelling . . . 29
  - 5.1 Dispersed Flows . . . 29
  - 5.2 Lift Force . . . 31
  - 5.3 Wall Lubrication Force . . . 34
  - 5.4 Drag Force . . . 36
  - 5.5 Turbulent Dispersion Force . . . 37
  - 5.6 Virtual Mass Force . . . 38
  - 5.7 Bubble Diameter Modelling . . . 39

5.8 Aspect Ratio Modelling . . . . . 41
5.9 Segregated Flows . . . . . 44

**3 Implementation in OpenFOAM 48**
1 twoPhaseEulerFoam . . . . . 48
2 reactingTwoPhaseEulerFoam . . . . . 49
2.1 The Species Transport Equation . . . . . 51
2.2 The Interface Composition Model . . . . . 53
3 PIMPLE Loop . . . . . 60
4 The Alpha Equation . . . . . 61
4.1 The Dilatation Rate Term . . . . . 61
4.2 The Alpha Equation Source Terms . . . . . 65
4.3 The Fluxes . . . . . 65
5 MULES . . . . . 67
5.1 ExplicitSolve . . . . . 68
5.2 Limit . . . . . 69
6 Discretised Momentum Equation . . . . . 72
6.1 P-U Coupling . . . . . 74
7 Discretised Energy Equation . . . . . 75

**4 Simulations 78**
1 Introduction to the Case-study . . . . . 78
2 Cases Setup . . . . . 79
2.1 Benchmark Quantities . . . . . 80
2.2 Error quantification . . . . . 81
3 Case 2 Results . . . . . 83
3.1 `interFoam` . . . . . 83
3.2 `twoPhaseEulerFoam` . . . . . 88
3.3 Comparison . . . . . 92
4 Case 1 Results . . . . . 97
4.1 `interFoam` . . . . . 97
4.2 `twoPhaseEulerFoam` . . . . . 101
4.3 Comparison . . . . . 105

**5 Closure 110**
1 Conclusions . . . . . 110
1.1 Cavitation . . . . . 110
1.2 Interface Tracking . . . . . 111
1.3 Slip Velocity . . . . . 112
2 Suggestions for Future Work . . . . . 113
2.1 Cavitation . . . . . 113

2.2 Interface Tracking . . . . . 114

**A The & Operator and *divU* derivation** **116**
1 Discretisation of PDEs . . . . . 116
2 the calculation of divU . . . . . 118
2.1 the "&" operator . . . . . 118
2.2 The derivation of "divU" . . . . . 120

**B Interface-Composition Models** **122**
1 Lift Coefficient Models . . . . . 122
1.1 Moraga Model . . . . . 122
1.2 Legendre-Magnaudet Model . . . . . 123
2 Wall Lubrication Models . . . . . 123
2.1 Antal Model: . . . . . 123
2.2 Tomiyama Model . . . . . 124
2.3 Frank Model . . . . . 124
3 Drag Models . . . . . 125
3.1 Tomiyama Model . . . . . 125
3.2 Ishii-Zuber Model . . . . . 125
4 Turbulent Dispersion Models . . . . . 126
4.1 Lopez de Bertodano Model . . . . . 126
4.2 Burns Model . . . . . 127
4.3 Gosman Model . . . . . 127

# List of Figures


1 Comparison of simulated liquid volume fraction iso-surface snapshots ($\alpha_{iso} = 0.5$) and instantaneous photographs of the experiment from [7] for $p_{in} = 0.22$, 0.25 and $0.28MPa$. . . . . II
2 Initial configuration and boundary conditions for the test cases. VI
3 Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions. . . . . VII
4 Comparison between Vof's (left) and Euler-Euler's (right) charachteristic benchmark values . . . . . VII
5 Comparison between bubble circularities at different mesh resolutions for both two-fluid method (a1) and VoF (b1). Comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a2) and VoF (b2). Comparison between bubble centroids at different mesh resolutions for both two-fluid method (a3) and VoF (b3). . . . . VIII
6 Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions. . . . VIII
7 Comparison between Vof's (left) and Euler-Euler's (right) charachteristic benchmark values . . . . . IX
8 Comparison between bubble circularities at different mesh resolutions for both two-fluid method (a1) and VoF (b1). Comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a2) and VoF (b2). Comparison between bubble centroids at different mesh resolutions for both two-fluid method (a3) and VoF (b3). . . . . IX

1.1 Model throttle flow path between the sapphire windows and Throttle geometry "U" in $300\mu m$ thick sheet steel [8] . . . . . 2
1.2 Experimental rig and detail on nozzle geometry of Sou's experiment [7, 9] . . . . . 3
1.3 Experimental vs VoF simulation liquid volume fraction as simulated by Sou et al. [9] . . . . . 4
1.4 Concept for 3-phase simulation by combination of Eulerian Eulerian and Volume of Fluid approaches. . . . . 4

1.5 Comparison of simulated liquid volume fraction iso-surface snapshots ($\alpha_{iso} = 0.5$) and instantaneous photographs of the experiment from [7] for $p_{in} = 0.22$, 0.25 and $0.28MPa$. . . . . 5
1.6 Scketches of flow regimes for two-phase flow in vertical pipes [26]. 6
1.7 Test section photographs of upward air-water flow regimes:(a) Bubbly, (b) Spherical cap, (c) Stable slug, (d) Unstable slug, (e) Semi-annular, (f) Annular [27]. . . . . 7
1.8 Shape regimes of fluid particles in unhindered gravitational motion through liquids [28]. . . . . 8
1.9 Terminal velocity as a function of diameter for air bubbles in water [28]. . . . . 9

2.1 Eulerian-Lagrangian approach schematic. . . . . 14
2.2 Eulerian-Eulerian approach schematic. . . . . 16
2.3 Eulerian-Lagrangian vs Eulerian-Eulerian. . . . . 17
2.4 Control volume ($V$) in Eulerian framework for single phase flow. 18
2.5 Control volume ($V$) in Eulerian framework for multi-phase flow. 20
2.6 Graphical representation of interfacial scale averaging closure framework for dispersed two-fluid systems . . . . . 30
2.7 Graphical representation of most important momentum transfer phenomena. . . . . 31
2.8 Lift force generated on a bubble due to the linear shear flow of the fluid around it. . . . . 32
2.9 Lift force on undistorted (a) and distorted (b) bubble in a shear flow . . . . . 34
2.10 Lift coefficient regimes as described by Tomiyama [40] . . . . . 34
2.11 Continuous phase lubrication layer between the wall and the dispersed phase, i.e. the gaseous phase . . . . . 35
2.12 Scheme of Wake Coalescence phenomenon. . . . . 40
2.13 Scheme of Turbulent Eddy Coalescence phenomenon . . . . . 41
2.14 Scheme of Turbulent Breakup phenomenon . . . . . 41
2.15 Aspect ratio for a spherical bubble (a) and for an elliptical bubble (b) . . . . . 42
2.16 Graphical representation of interfacial resolving closure framework for segregated two-fluid systems . . . . . 44

3.1 Flow chart of *twoPhaseEulerFoam* solver as implemented in OpenFOAM-dev . . . . . 49
3.2 Flow chart of *reactingTwoPhaseEulerFoam* solver as implemented in OpenFOAM-dev . . . . . 50
3.3 Interface species transfer with phase change . . . . . 55

3.4 Phase change of dispersed phase . . . . . 57
3.5 OpenFOAM procedure to determine phase change and mass transfer. . . . . 58

4.1 Initial configuration and boundary conditions for the test cases. 80
4.2 Contour plot for $h = 1/640$. . . . . 83
4.3 Contour plot at time $t = 3$ for different mesh refinements $h$. . 84
4.4 Circularity at different mesh resolutions for the VoF methodology (a), focus on bubble contours at incipient bubble breakup ($t = 2.6$)(b). . . . . 85
4.5 Centroid position over time (a), mean rise velocity (b) calculated for VoF method at different mesh resolutions. . . . . 86
4.6 Contour plot for $h = 1/640$. . . . . 88
4.7 Contour plot at time $t = 3$ for different mesh refinements $h$. . 89
4.8 Circularity at different mesh resolutions for the two fluid methodology (a), focus on bubble contours at incipient bubble breakup ($t = 2.2$)(b). . . . . 90
4.9 Bubble circularity evolution against time (a) for different mesh resolutions and bubble contours at $t = 2.2$ for two fine resolutions (b) . . . . . 91
4.10 Centroid position over time (a), mean rise velocity (b) calculated for two fluid method at different mesh resolutions. . . . 91
4.11 Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions. . . . 93
4.12 comparison between bubble circularities at different mesh resolutions for both two-fluid method (a) and VoF (b). . . . . 94
4.13 comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a) and VoF (b). . . . . 95
4.14 comparison between bubble centroids at different mesh resolutions for both two-fluid method (a) and VoF (b). . . . . 96
4.15 Visual confrontation of contours at last time-step for different mesh refinement . . . . . 97
4.16 bubble circularity over time (a), zoom of bubble circularity in the second half of bubble evolution (b) calculated for VoF method at different mesh resolutions. . . . . 98
4.17 Centroid position over time (a), mean rise velocity (b) calculated for VoF method at different mesh resolutions. . . . . 99
4.18 Visual confrontation of contours at last time-step for different mesh refinements. . . . . 101
4.19 bubble circularity over time calculated for two-fluid method at different mesh resolutions. . . . . 102

4.20 Centroid position over time (a), mean rise velocity (b) calculated for VoF method at different mesh resolutions. . . . . 103
4.21 Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions. . . . 105
4.22 comparison between bubble circularities at different mesh resolutions for both two-fluid method (a) and VoF (b). . . . . 106
4.23 comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a) and VoF (b). . . . . 107
4.24 comparison between bubble centroids at different mesh resolutions for both two-fluid method (a) and VoF (b). . . . . 108

5.1 Different barotropic models for water at T=293 K . . . . . 113

A.1 Example matrix. . . . . 117
A.2 "X" refears to diagonal elements, "O" refears to owner cells and "N" refears to neighbouring cells . . . . . 117
A.3 The H operator . . . . . 120
A.4 Schematic representation of *divU* derivation . . . . . 121

# List of Tables


1 Pysical parameters defining the test cases . . . . . VI

2.1 Coefficients for the generic transport equation for single-phase flow. . . . . 20
2.2 Multiplier terms for Gosman et al. [39] . . . . . 31
2.3 Summary of coefficients for source terms in interfacial area transport equation according to Ishii and Hibiki [43] . . . . . 42

4.1 Pysical parameters defining the test cases . . . . . 80
4.2 Minimum circularity and maximum rise velocities with corresponding time and centroid final position . . . . . 86
4.3 Minimum circularity and maximum rise velocities with corresponding time and centroid final position . . . . . 92
4.4 Minimum circularity and maximum rise velocity with corresponding time and centroid final position . . . . . 99
4.5 Relative error norms and convergence orders for VoF. . . . . 100
4.6 Minimum circularity and maximum rise velocity with corresponding time and centroid final position . . . . . 103
4.7 Relative error norms and convergence orders for VoF. . . . . 104

# Listings


3.1 YEqns.H file . . . . . 51
3.2 Species transport equation as defined in the "BasePhaseModel" 52
3.3 Template of an interfaceComposition call in phaseProperties . 53
3.4 The phase dilatation rate in `twoPhaseSystem.C` . . . . . 61
3.5 Definition of *divU* in `pEqn.H` . . . . . 62
3.6 Compressible part of the pressure equation for phase 1 in `pEqn.H` 62
3.7 The compressibility contribution due to mass transfer to pressure and alpha equation . . . . . 64
3.8 *MULES::explicitSolve* call in `twoPhaseSystem.C` . . . . . 68
3.9 Momentum equation for two-phase systems in OpenFOAM as formulated in file `UEqns.H` located in **../multiphase/reactingEulerFoam/reactingTwoPhaseEulerFoam/pU** . . 73
3.10 The discretized momentum equation as implemented in the basePhaseModel . . . . . 73
3.11 Energy equation for two-phase systems as formulated in file `EEqns.H` located in: **application/solvers/multiphase/reactingEulerFoam/reactingTwoPhaseEulerFoam** . . . . . 75
3.12 Energy equation for a two-phase system in OpenFOAM. . . . 76

# Chapter 1

# Introduction

## 1 Background

Multiphase flows are adopted in several industrial applications such as power plants [1, 2], oil and gas field management [3], combustion and in general chemical processes [4], cavitation in technical applications [5, 6]. The study of multiphase flows and in particular of two-phase fluids, which will represent the main subject of this thesis, shows a large amount of variants that are worth being discussed. Two-Phase flows can be categorized either by the physical state of the constituents (gas-solid, gas-liquid, solid-liquid, liquid-liquid flows) if immiscible liquids are considered or by the topology of the interface (segregated vs dispersed flows).

The direct measurement of characteristic flow variables such as velocity, volume fraction and temperature are not always possible mainly because the process could be altered by the acquisition method or because the equipment to be tested is too small to fit any experimental system of measurement, as in the case of the study of internal nozzle flows in fuel injectors. Flow inside single orifice nozzles and fuel injectors with complex geometries are gaining significant attention from researchers, as evidenced by the several conferences organized yearly around the world on this topic. This interest in the flow characteristics of nozzles and injectors is largely due to their use in several engineering applications, the most important being in the automotive industry. For instance, cavitation occurring in single orifice nozzles and fuel injectors plays a pivotal role in fuel plume atomization required for better fuel economy and reduction of emissions (soot and particulate matter). However, cavitation also can lead to a significant reduction in the nozzle volumetric efficiency (in diesel engines), in the spray direction stability and to a potential

damage of the hardware components, thus contributing to increased cost and reduced reliability of these injectors. These conflicting attributes has spurred a renewed interest in understanding the complicated flow-physics inside these devices [5].

As mentioned before obtaining experimental evidences from this kind of processes is really cumbersome and most of the time includes the use of pilot studies (experiments carried out in thermo-hydrodynamic similitude with the case study) [7–9] as well as empirical correlations. Winklhofer et al. [8] proposed a 2D experimental rig to capture water cavitation in different nozzle geometry configurations at different pressures. They used back illumination and imaging of the flow model with a CCD camera in order to capture and visualize the cavitation field. The CDD camera allowed to capture pressure and density field as well, through a Mach Zehnder interferometer arrangement [8], while velocity profiles were calculated with a fluorescence tracing method.

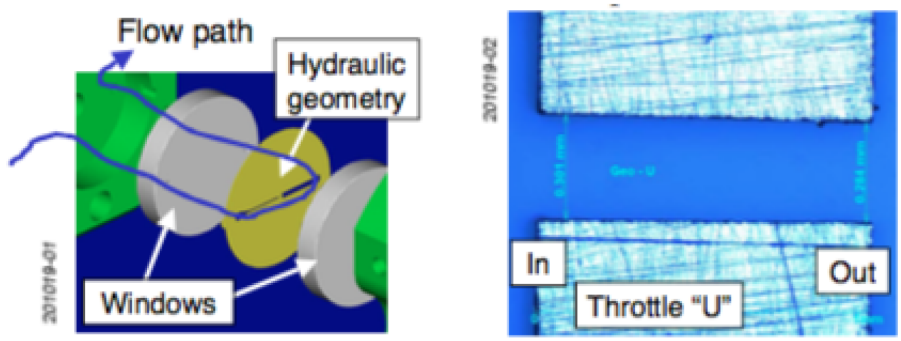


Figure 1.1: Model throttle flow path between the sapphire windows and Throttle geometry "U" in 300$\mu m$ thick sheet steel [8]

Today, CFD simulations of internal nozzle flows are becoming very popular and are performed in combination to experimental studies, to expand the applicability of the solution method to a wider spectrum.

Between the several CFD methodologies available in the literature, the VoF method (Volume of Fluid) is preferred for simulations of injectors [7, 9–12]. Main drawback of this methodology is the lack of information about the two fluids since a single momentum equation is solved, meaning that the phase slip velocity is modelled giving a low quality tracking of the phases'

velocities, on the other hand they allow a correct modelling of the phase-interface that is really important to track in cavitation problems since the phase exchange happens at the interface.

With an experimental set-up similar to the one used by Winklhofer et al. [8], Morozov and Iben [12] in 2008 presented the validation of a cavitation model based on Rayleigh-Plesset equation by the software ANSYS CFX-11.0. The validation was carried on with the assumption of single velocity and pressure between the two phases, i.e. a VoF model was tested.
More recently Bicer, Sou and Tomiyama [7, 9] simulated incipient cavitation in a more injector-like geometry by the VoF method. Following in image 1.2 are reported the experimental rig and a detail on the nozzle geometry. They were able to obtain with a VoF method good resemblance with the

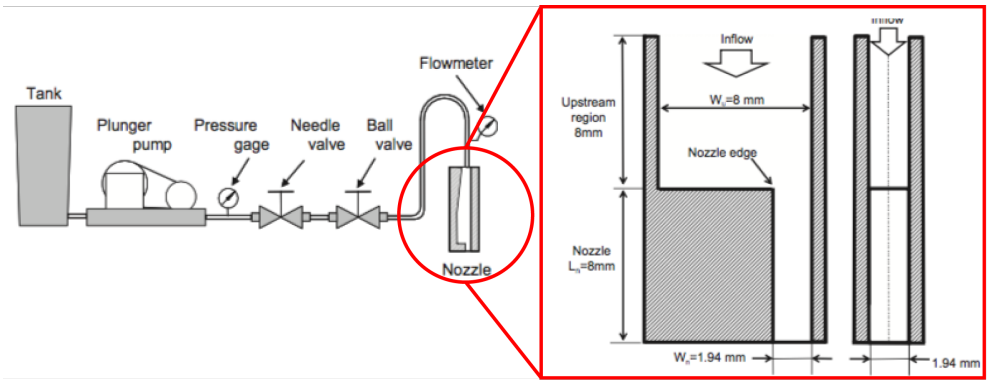


Figure 1.2: Experimental rig and detail on nozzle geometry of Sou's experiment [7, 9]

experimental evidences as reported in figure 1.3

Only in 2016 Edelbauer [10] proposed the validation of an hybrid solver: a three-phase system consisting of liquid, vapour and gas is applied for the Volume of Fluid simulation of the liquid disintegration in order to resolve the liquid-gas interface. The method presented by Edelbauer overcomes the problem of high computational effort needed by surface-capturing methods (each bubble is resolved singularly) by modelling the liquid-vapour interface within the Eulerian Eulerian framework, while the liquid-gas interface is described by the VoF method as graphically summarized in image 1.4.

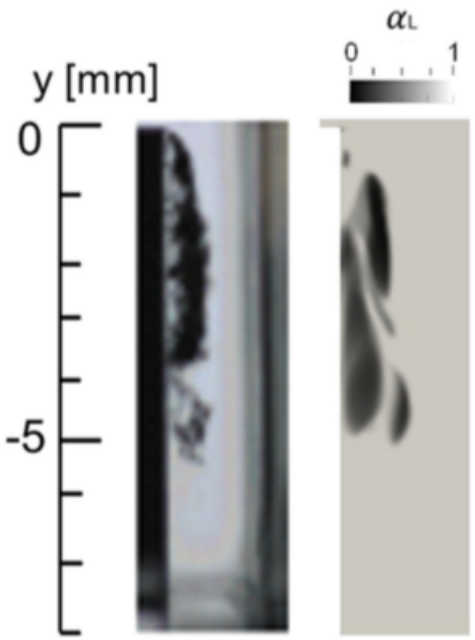


Figure 1.3: Experimental vs VoF simulation liquid volume fraction as simulated by Sou et al. [9]

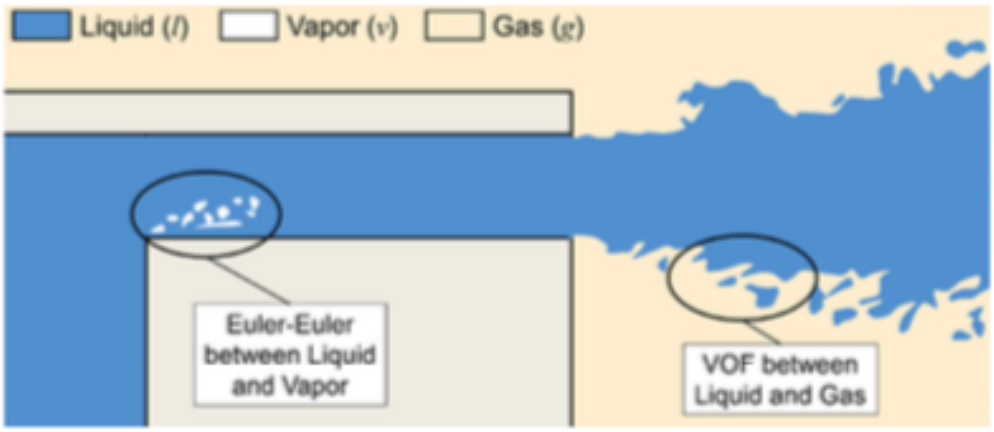


Figure 1.4: Concept for 3-phase simulation by combination of Eulerian Eulerian and Volume of Fluid approaches.

With this approach Edelbauer was able to reach (with reasonable computational effort) a more accurate solution with respect to Sou's one. Some results are plotted in figure 1.5 and show that the vapour fraction, even at small pressure jump, is intercepted more accurately.

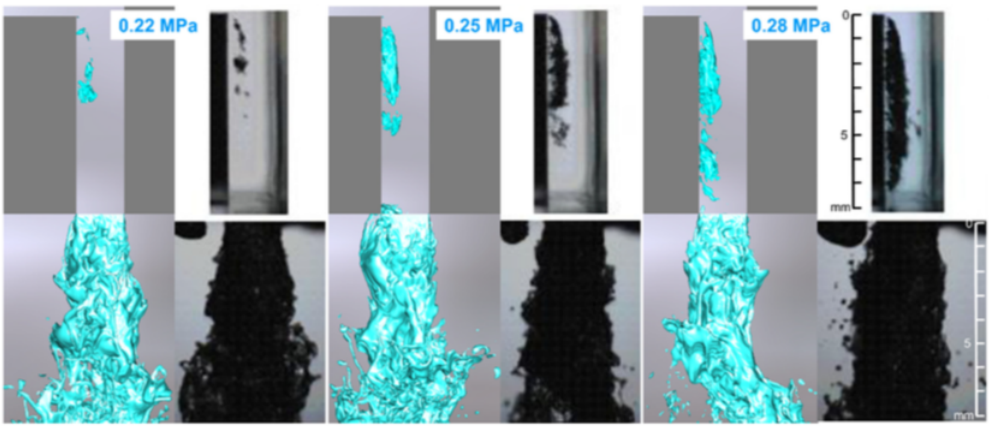


Figure 1.5: Comparison of simulated liquid volume fraction iso-surface snapshots ($\alpha_{iso} = 0.5$) and instantaneous photographs of the experiment from [7] for $p_{in} = 0.22$, 0.25 and $0.28 MPa$.

As it is possible to infer from what said so far, in order to capture accurately the cavitation during injection, two fundamental characteristics are requested from a solver:

- Capacity to capture the gas/liquid interface;
- Accurately take into account slip velocity between phases.

In order to better understand how the multiphase flow can manifest itself, and hence how the interface should be transported in different flow situation a brief description and classification of multiphase flows and bubble dynamics is presented in the following section.

# 2 Multiphase-Flow Dynamics

A multiphase flow is the flow of a mixture of phases such as gases (bubbles) in a liquid, or liquid (droplets) in gases, and so on. Dispersed phase flows are flows in which one phase consists of discrete particulates, such as droplets in a gas or bubbles in a liquid. On the other hand, separated/segregated flows happens when the two phases are separated by a line of contact i.e. the interface. In this section, the different flow regimes are analysed through the study of a two-phase fluid (gas+liquid) in a vertical pipe [22–25]. Depending on the combination of gas and liquid superficial velocities, a specific flow regime may occur in vertical upward gas-liquid flow. We are primarily interested in four clearly distinguishable flow regimes, namely: bubbly, slug,

churn and annular flows, which will be referred to as distinct regimes. For vertical up-flow, as the amount of gas is gradually increased, the following four distinct regimes evolve:

- **Bubbly Flow:** In this configuration, there is a continuous liquid phase and the gas phase is dispersed as bubbles within the liquid. The bubbles travel with a complex motion within the flow, may be coalescing and are generally of non-uniform size.
- **Slug flow:** Characteristic bullet-shaped bubbles, often called Taylor bubbles, flow up the pipe surrounded by a thin film of liquid. The liquid slug body between the Taylor bubbles often contains a dispersion of smaller ones.
- **Churn flow:** At higher velocities, the Taylor bubbles in slug flow break down into an unstable pattern in which there is an oscillatory motion or churning of liquid in the tube.
- **Annular flow:** This configuration is characterized by liquid travelling as a film on the channel walls. Part of the liquid can also be carried as drops in the central gas core.

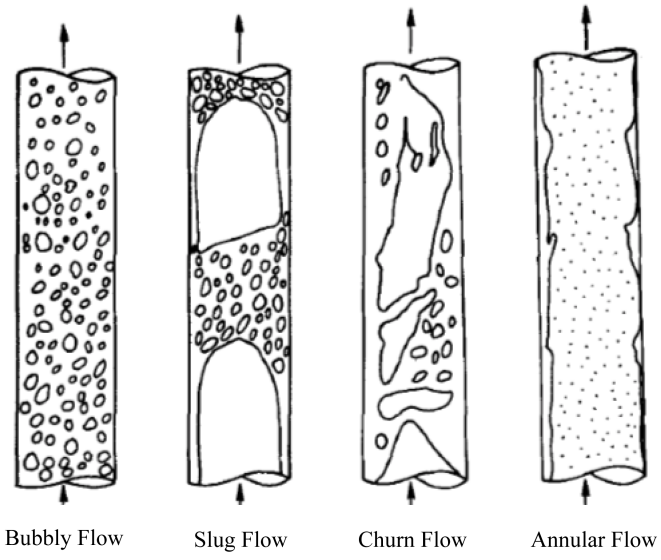


Figure 1.6: Scketches of flow regimes for two-phase flow in vertical pipes [26].

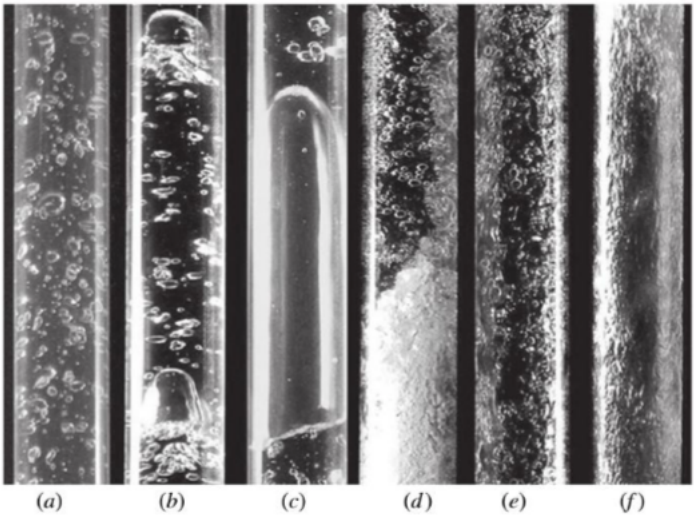


Figure 1.7: Test section photographs of upward air-water flow regimes:(a) Bubbly, (b) Spherical cap, (c) Stable slug, (d) Unstable slug, (e) Semi-annular, (f) Annular [27].

Figures 1.6 and 1.7 present respectively a schematic and visual representation of the different flow regimes.

This work will focus mainly on the bubbly flow regime since cavitation in injectors, as highlighted in the previous chapter, fits better this case. Following a description of the bubble shape in regime (a) and (b) of figure 1.7 is presented.

## 2.1 Bubble Motion

The bubbles dispersed in a fluid flow doesn't behave like solid particles (fluidized beds). Firstly the momentum transfer from the continuous fluid phase results in a circulation of the fluid inside the bubble and secondly the shape of the bubble is not fixed. This has to be taken into account when modelling drag and lift forces between the phases and it will be described in section 5 of chapter 2 .
Two dimensionless numbers can be defined to describe the bubble shape:

$$\mathcal{E}o = \frac{g_{eff}|\Delta\rho|d^2}{\sigma} = \frac{\text{Buoyancy Force}}{\text{Surface Tension}} \tag{1.1}$$

$$\mathcal{M}o = \frac{g_{eff}\mu_c^4|\Delta\rho|}{\rho_c^2\sigma^3} \tag{1.2}$$

Where $\mathcal{E}o$ stands for Eötvös number and $\mathcal{M}o$ stands for Morton number. $\sigma$ is the surface tension and $\Delta\rho$ is the density difference between continuous and dispersed phase, i.e $\Delta\rho = \rho_d - \rho_c$. $g_eff$ is the magnitude of the effective acceleration $|\vec{g}_{eff}| = |\vec{g} - \frac{D\vec{U}_c}{Dt}|$, the effective acceleration takes into account the acceleration of the continuous phase that can be appreciable. Figure 1.8

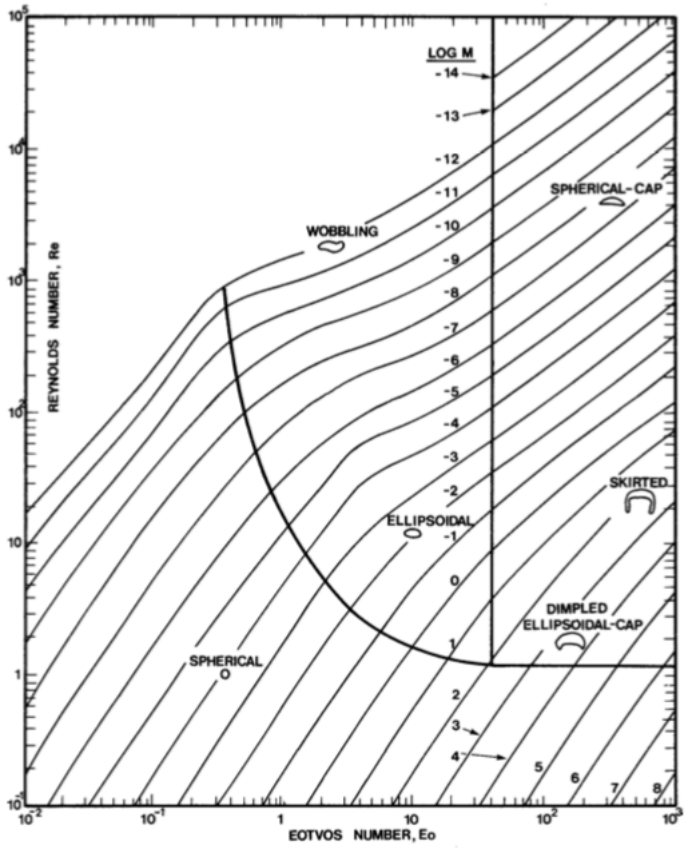


Figure 1.8: Shape regimes of fluid particles in unhindered gravitational motion through liquids [28].

shows the characteristic shapes of fluid particles rising in a quiescent fluid as function of Morton and Eötvös numbers. The diagram provides also the Reynolds number based on the terminal velocity as function of the same parameters. For low $\mathcal{E}o$ spherical shape is retained while for larger ones the bubble takes an ellipsoidal shape or largely deformed. This behaviour is more pronounced for low $\mathcal{M}o$ since the continuous phase has a low viscosity, i.e when a fluid particle rises through a continuous phase, work is done on the continuous phase at a rate equal to the rise velocity times the buoyancy force acting on the fluid particle. In very viscous fluids the energy is dissipated

mostly by laminar viscous dissipation but with a low viscosity continuous phase some of it is dissipated through wake shedding. Which is responsible for the secondary motion of the bubble that is the superposition of two types of motion: oscillation on the bubble trajectory and bubble shape oscillations. Figure 1.9 highlights the terminal rise velocity of air bubbles in water as a function of the bubble diameter. Three regimes are identified: spherical, ellipsoidal and spherical-cap. The shaded area represents the dependence of the bubble rise velocity also on the presence of surfactants and as it is possible to see from the figure this dependence mainly affects ellipsoidal bubbles. These features will be recalled in chapter 4 where a numerical simulation for the solver validation will be carried on. It will be possible to see the transition of the bubble shape between the different regimes as well as verify the velocity distribution as function of the bubble shape.

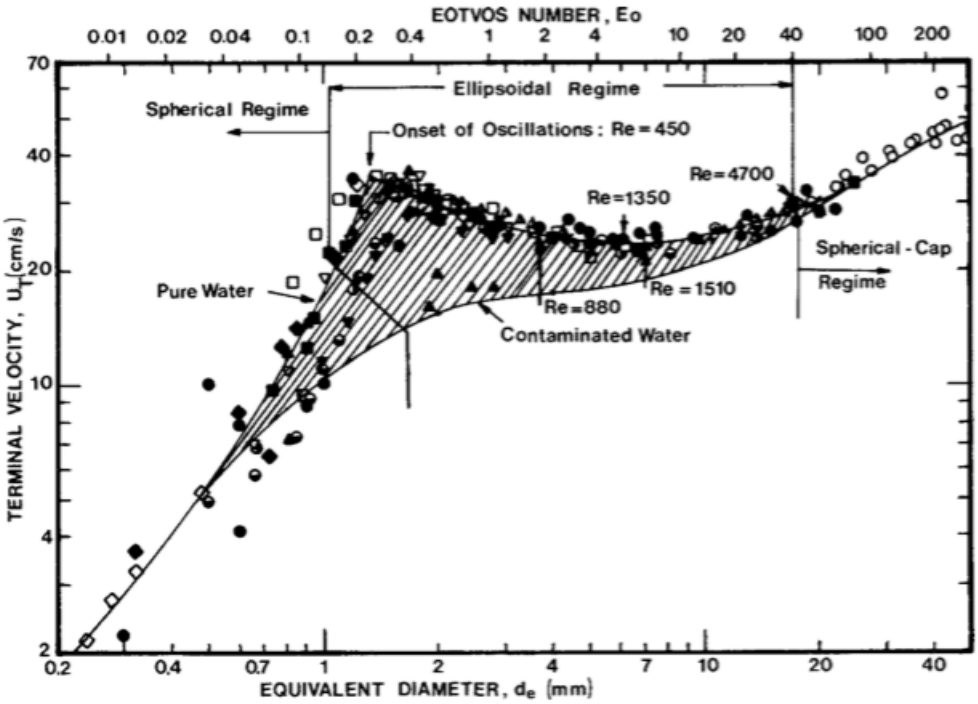


Figure 1.9: Terminal velocity as a function of diameter for air bubbles in water [28].

# 3 Objectives

The aim of this work is then to evaluate the feasibility of the application of an Eulerian-Eulerian solver to simulate cavitation in injector-like geometries. In order to achieve this goal, the research started with the selection of the most suitable solver already implemented in OpenFOAM. Among the different solvers based on the two-fluds Eulerian-Eulerian approach available in OpenFOAM, `reactingTwoPhaseEulerFoam` was chosen: the reason is that this solver, originally designed for multiphase chemical reactors and fluidized-bed combustors, is able to track the mass fraction of the chemical species and to account for chemical reactions. The original idea was to implement a dummy chemical reaction in order to "emulate" cavitation in the system by an homogeneous reactor model. With this in mind, most of the efforts were oriented to obtain a full description of the solver implementation and to outline differences, when present, with other solvers from the existing literature [2, 15, 19–21]. In the meanwhile, the interconnection between the species transport equations and the Eulerian-Eulerian framework transport equations was investigated, with particular focus on the correlation between the mass source term and the transport of the volume (void) fraction. The scope of this investigation was to understand whether it is possible to correlate mass transport and volume fraction transport with a cavitation source term. The findings about this will be presented as a part of chapter 3 about OpenFOAM implementation of two-fluid model.

Additionally, the metodology employed in OpenFOAM for interface tracking in multiphase solvers was studied. A numerical experiment was performed to check whether the interface was being transported correctly. Finally, to give a quantitative evaluation a complete analysis on benchmark quantities defined in [13] was performed on the Eulerian-Eulerian solver and compared with the VoF solver `interFoam` that, as shown before [7, 9], was proven as an efficient method for interface tracking. The study about interface transport wasn't however limited to the results of the benchmark tests, but a physical description of interface tracking in two-phase flows, as well as comparison of the solver implementation with literature [19] was performed. Results about interface tracking are described in chapter 2 and 4, respectively in two-fluid model description and simulations.

# 4 Thesis Outline

The remainder of this thesis is structured as follows:

**Chapter 2:** CFD methodologies for two-phase flows and generalized transport equations are derived. The two-fluid model for both cavitating and non cavitating systems are derived as well as the inter-phase momentum transfer terms for both dispersed and segregated flows;

**Chapter 3:** The `reactingTwoPhaseEulerFoam` solver is described in detail highlighting the equations and models described in chapter 3 as well as the link between species transport and phase fraction transport equations;

**Chapter 4:** The benchmark case is performed for both Euler-Euler and VoF methodologies at different mesh resolutions and then the results are compared;

**Chapter 5:** The main findings and conclusions are summarized. Issues and suggestions for future research are presented.

# Chapter 2

# Two-Fluid Model

In this Chapter, an introduction to the classification of Multiphase flows, as well as to bubble shape and dynamics, is discussed. Two different CFD methodologies for two-phase flow are compared and the general conservation equations are derived. Then, the two-fluid model for both non-cavitating and cavitating systems are described. The governing equations for compressible fluids are listed along with turbulence and inter-phase momentum transfer modelling for dispersed flows and segregated ones.

## 1 CFD Metodologies for Two-Phase Flow

The present study needs a model to track the velocities of both phases separately. This is why a volume of fluid (VoF) methodology was discarded at the beginning of this work. Being a free-surface modelling technique it allows correct tracking of the fluid interface but, the Navier-Stokes equation solved alongside the void fraction transport equation are referred to the mixture velocity instead of tracking the two separate velocities of the two fluids. This is why, for a more accurate analysis either an *Eulerian-Lagrangian* or an *Eulerian-Eulerian* method has to be used. In the following sections the two methodologies will be briefly described to justify the choice of the solver used in this thesis.

### 1.1 Eulerian-Lagrangian Approach

In the *Eulerian-Lagrangian* approach the two phases are referred to as the continuous and the dispersed phase. A macroscopic description of the dispersed phase is obtained by replacing the microscopic conservation equations with a discrete formulation. In this discrete formulation, the dispersed phase

is represented by individual droplets, which are tracked through the flow domain by solving an appropriate equation of motion. The equation of motion is the conservation equation of momentum expressed in the Lagrangian formulation, in which the dependent variables are the properties of material particles that are followed in their motion. On the other hand, the conservation equations for the continuous phase are expressed in the Eulerian frame, where the fluid properties are considered as functions of space and time in an absolute (or inertial) frame of reference [15].

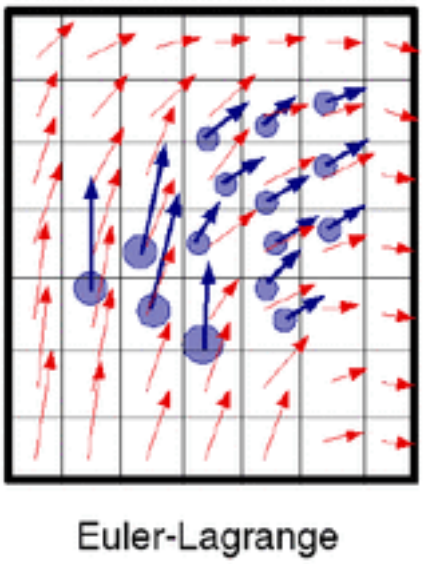


Figure 2.1: Eulerian-Lagrangian approach schematic.

As it is depicted in figure 2.1 it is possible to distinguish the continuous phase, modelled through an Eulerian framework, through the red arrows (each cell of the control volume has a velocity vector for the continuous phase). On the other hand the dispersed phase, modelled through a Lagrangian framework, is represented through the blue arrows, which highlights that each single droplet of dispersed phase is tracked singularly.

For sufficiently dilute suspensions, where the particle size is small, the influence of the dispersed phase on the motion of the continuous phase can be neglected. The coupling between the phases is then said to be one-way. However, the matter is somewhat complicated if the motions of the continuous and the dispersed phase are closely coupled, i.e. the continuous phase influences the motion of the DPEs (Dispersed Phase Elements) and vice versa. This two-way coupling can be taken into account in the Eulerian-Lagrangian

model with relative ease and is done by accounting for the influence of the dispersed phase in the momentum equation and the turbulence model of the continuous phase.

An important advantage of this methodology lies in the possibility to store information about each single element of dispersed phase, these information will then be accounted for in each equation of motion, solved separately for each droplet. On the other hand, problems arise when the dispersed phase fraction is high. The first problem is that the computational effort required for the dispersed phase is proportional to the number of DPEs because each DPE requires the solution of its equations of motion. The problem can be circumvented by calculating the motion of a finite number of computational parcels, each containing DPEs possessing the same characteristics such as size, velocity, shape etc.. However, the number of DPEs in a parcel may have a marked effect on some of the obtained statistics, as shown by [29], [30]. Secondly, for moderate to high phase fractions, the increased coupling between the DPEs and the continuous phase can introduce numerical stability problems [30], which are difficult to handle in a mixed Euler/Lagrange framework [15].

## 1.2 Eulerian-Eulerian Approach

Here both phases are described through an Eulerian framework. Each phase is treated as a continuum, each inter-penetrating each other, and is represented by averaged conservation equations. The averaging process introduces the phase fraction $\alpha$ into the equation set, which is defined as the probability that a certain phase is present at a certain point in space and time [15].

As did before for the Eulerian-Lagrangian model in figure 2.2 is depicted a schematic of the Eulerian-Eularian model. Here, as highlighted before, for each cell two sets of vectors are present the red ones representing the continuous phase while the blue one represents the dispersed one. Each cell presents also an $\alpha$ weighting value which constitute the dispersed phase fraction and allows to identify where the dispersed phase droplets are most probably located in the control volume.

Due to the loss of information associated with the averaging process, additional terms appear in the averaged momentum equation for each phase, which require closure. In addition to the Reynolds stresses, which enter into the averaged single-phase flow equations, an extra term that accounts for the transfer of momentum between the phases appears. This term is known as the averaged inter-phase momentum transfer term and accounts for the

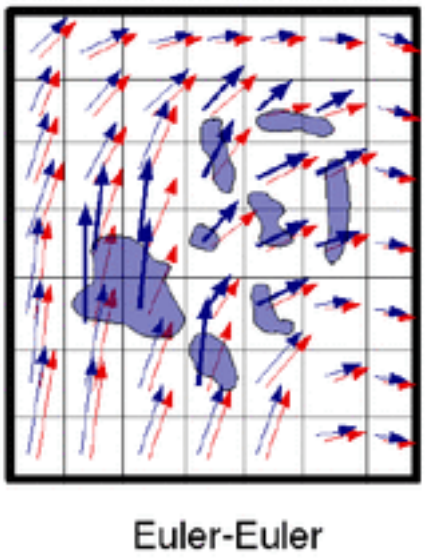


Figure 2.2: Eulerian-Eulerian approach schematic.

average effect of the forces acting at the interface between continuous phase and the DPEs. The two-fluid methodology is applicable to all flow regimes, including separated, dispersed or intermediate regimes, since the topology of the flow is not prescribed. However, the formulation of the inter-phase momentum transfer term and the two-phase turbulence model is the crux of the two-fluid methodology because it depends on the exact nature of the flow. Consequently, the resulting predictive capabilities rely heavily on them. The derivation of the average inter-phase momentum transfer term starts from instantaneous momentum transfer term. The main components of this term are due to the drag, lift and virtual mass forces acting at the interface between the two phases, and will be deeply analysed in chapter 2 .

It should be noted that the Eulerian-Eulerian model, by definition, incorporates two-way coupling (interphase momentum exchange) and, despite the other complexities, is the preferred method for engineering applications, especially when the phase fraction is high. For these reasons it is adopted in this study. This choice mainly relies on the high computational costs of the Eulerian-Lagrangian method despite the need of models for the Eulerian-Eulerian model, as showed in figure 2.3 and the complications that arise at high phase fractions, namely the need of two-way coupling.

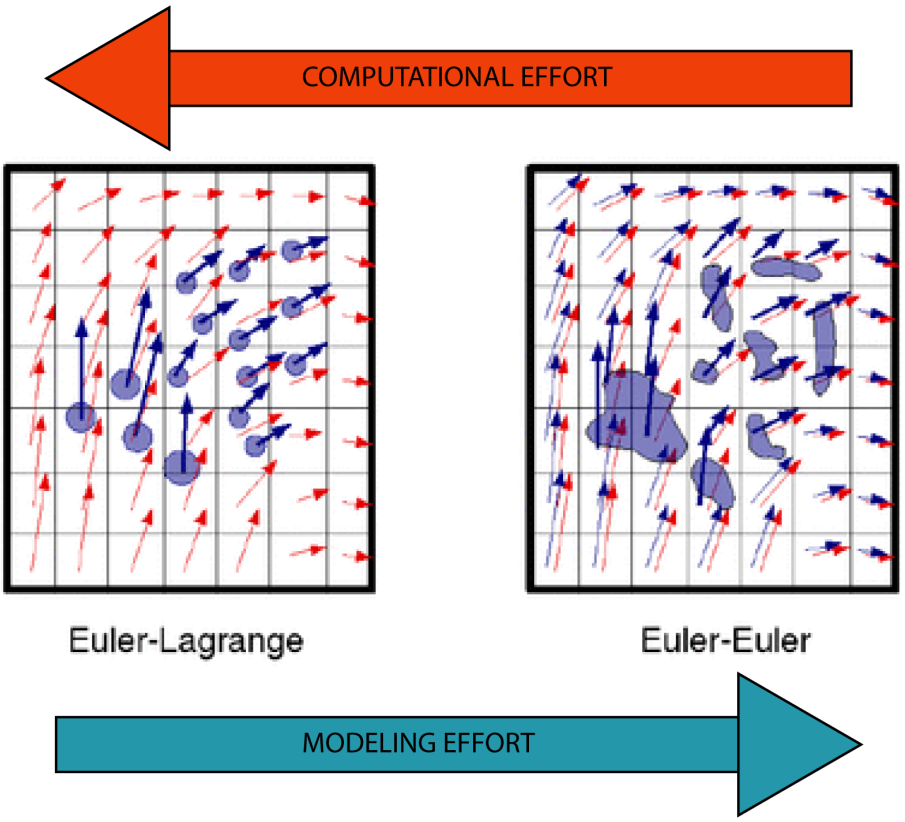


Figure 2.3: Eulerian-Lagrangian vs Eulerian-Eulerian.

# 2 Local Instantaneous Bulk Conservation Equations

Here, the generalized transport equation for both single phase and two phase flows is derived. This procedure will provide the reader with the basic transport equations that will be presented in the following chapters.

## 2.1 Single-Phase Flow

Let's consider a control volume $V$, spatially fixed within an Eulerian framework as illustrated in figure 2.4 . Over this control volume the balances of a generic property $\phi$ will be formulated. It represents the intensive equivalent of extensive balance quantities such as mass, momentum and energy. To do so, let's suppose that a fluid of density $\rho$ crosses the control volume's surface

$S$ at an infinitesimal surface element $dS$ with $\vec{U}$ velocity.

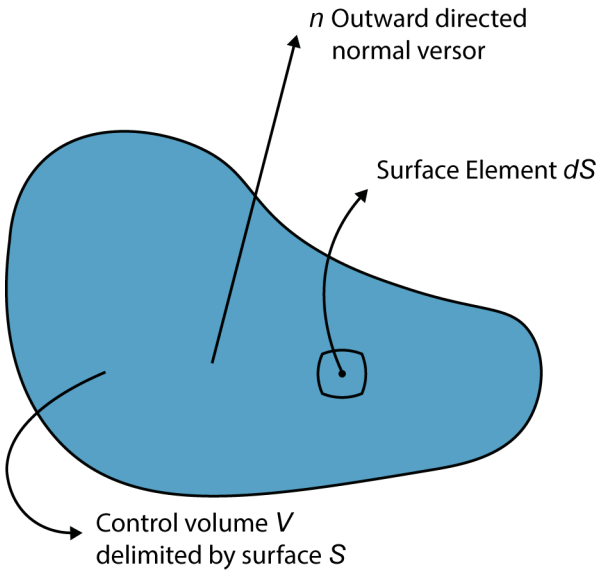


Figure 2.4: Control volume ($V$) in Eulerian framework for single phase flow.

The transport of $\Phi$ across $V$ is generally constituted from different contributions that represents either surface or volume contributions as follows [16, 17]:

$$\begin{aligned} \frac{d}{dt}\int_{V(t)} \rho\Phi \quad dV = &- \oint_{S(t)} \vec{n}\cdot(\rho\phi\vec{U}) \quad dS \\ &- \oint_{S(t)} \vec{n}\cdot(-\Gamma_{\Phi,d}\nabla\Phi) \quad dS \\ &+ \int_{V(t)} S_\Phi(\Phi) \quad dV \end{aligned} \tag{2.1}$$

Where $\Gamma_{\phi,d}$ represents the generic transport coefficient for the diffusive transport of quantity $\phi$ and $S_\phi$ denotes the generic volumetric source term. The transport equation (2.1) states that the total amount of $\phi$ will change within the control volume $V$ if a diffusive or convective net flux across the bounding control surface $S$ takes place, or a source/sink within the control volume $V$ itself causes an increase/decrease of $\phi$ .

Even tough from equation (2.1) it was possible to identify the different contributions to property $\phi$ change inside the control volume, this doesn't represent the final formulation of the generalized transport equation, in order

to obtain that Leibnitz' and Gauss' theorems has to be introduced to allow a simplification of the formula:

**Leibnitz' Theorem [31]:**

$$\frac{d}{dt}\int_{V(t)} \vec{a} \quad dV = \int_{V(t)} \frac{\partial \vec{a}}{\partial t} \quad dV + \oint_{S(t)} \vec{a}\left(\vec{n}\cdot\vec{U}_S\right) \quad dS \tag{2.2}$$

Where $\vec{a}$ is an arbitrary vector and $\vec{U}_S$ is the velocity of the bounding surface $S$ that is null and hence the second term can be neglected, meaning that Leibnitz' theorem allows the transport of the derivative inside the integral operator.

**Gauss' Theorem [31–33]:**

$$\int_{V(t)} \nabla\cdot\vec{a} \quad dV = \oint_{S(t)} \vec{n}\cdot\vec{a} \quad dS \tag{2.3}$$

$$\int_{V(t)} \nabla\vec{a} \quad dV = \oint_{S(t)} \vec{n}\vec{a} \quad dS \tag{2.4}$$

This theorem allows to pass easily from a surface integral to a volume one.

Applying now Libinitz' theorem (Eq. (2.2)) to the L.H.S. of equation 2.1 and Gauss' Theorem (Eqns. (2.3) and (2.4)) to the R.H.S. of the same equation it is possible to derived a simplified version of the generalized transport equation:

$$\int_{V(t)} \left[\frac{\partial}{\partial t}(\rho\Phi) + \nabla\cdot(\rho\Phi\vec{U}) - \nabla\cdot(\Gamma_{\Phi,d}\nabla\Phi) - S_\Phi(\Phi)\right] \quad dV = 0 \tag{2.5}$$

From (2.5) it is possible to extract the generic differential transport equation deleting the dependence on the control volume ($V$) and hence extracting the square brackets:

$$\frac{\partial}{\partial t}(\rho\Phi) + \nabla\cdot(\rho\Phi\vec{U}) - \nabla\cdot(\Gamma_{\Phi,d}\nabla\Phi) - S_\Phi(\Phi) = 0 \tag{2.6}$$

After This demonstration it results easier the derivation of the various transport equations that will be solved in the following chapters by the OpenFOAM solvers. To give an idea to the reader mass, momentum and species transport equations can be derived substituting the coefficients of table 2.1 into equation (2.6), where $\mu$ is the kinematic viscosity (deriving from Newton's law), $D_i$ is the species diffusivity (deriving from Fick's law) and $\hat{R}_i$

Table 2.1: Coefficients for the generic transport equation for single-phase flow.

| Balance | $\Phi$ | $\Gamma_{\phi,d}$ | $S_\phi$ |
|---|---|---|---|
| Mass | 1 | 0 | 0 |
| Momentum | $\vec{U}$ | $\mu$ | $\rho\vec{g} - \nabla p$ |
| Species | $x_i$ | $\rho D_i$ | $\hat{R}_i$ |

is the reaction rate relative to specie $i$. The energy equation is a little bit trickier due to its various formulations, in this work we will refer to the formulation in chapter 2 section 3.3 . The derivation is similar to the other transport equations and it is recommended to consult [16, 17, 19] for a more detailed derivation of the transport equations.

## 2.2 Two-Phase Flow

We want now to extend the process explained in the previous section to a two-phase and in general to a multi-phase system. The control volume has the same properties as above, but it is now composed by several phase volumes ($V_k \subseteq V$), each possessing an interface separating it from the neighbouring phase $S_I$. As a consequence of that the control surface $S$ is composed of phase surfaces ($S_k \subseteq S$).

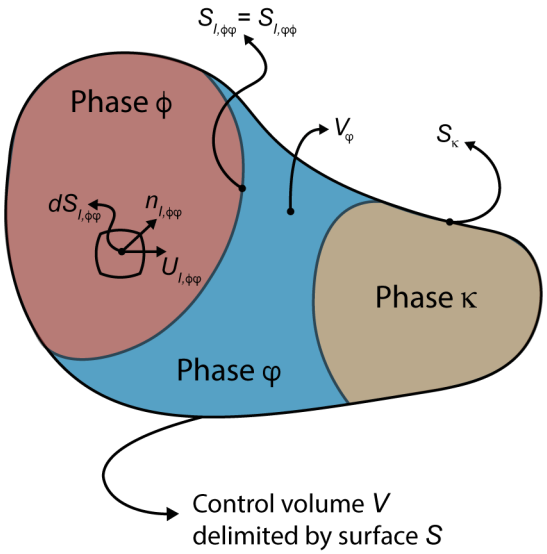


Figure 2.5: Control volume ($V$) in Eulerian framework for multi-phase flow.

This is schematically represented in figure 2.5, in which are also repre-

sented:

- $dS_{I,\phi\varphi}$: Interface element;
- $\vec{U}_{I,\phi\varphi}$: Interface velocity between phase $\phi$ and $\varphi$;
- $\vec{n}_{I,\phi\varphi}$: Outward phase unit normal vector.

In order to take into account the multi-phase effect on the transport equation of generic quantity $\Phi$ the balances will be considered on phase surfaces $S_k$ and phase volumes $V_k$.
Examining the transport of $\Phi$ within the control volume $V$ , one has to bear in mind that due to the presence of two phases, interfacial contributions have to be taken into account that stem from both phases $\varphi$ and $\phi$. As will be seen, these in effect couple the bulk phase transport equations derived in the antecedent of this section. The overall balance equation for $\Phi$ within the control volume V holding two phases $\varphi$ and $\phi$ reads:

$$\begin{aligned}\sum_{k=\varphi,\phi}\left(\frac{d}{dt}\int_{V_k(t)}\rho\Phi \quad dV\right) = &-\sum_{k=\varphi,\phi}\int_{S_k(t)}\vec{n}_k\cdot(\rho\Phi\vec{U}) \quad dS\\ &-\sum_{k=\varphi,\phi}\int_{S_k(t)}\vec{n}_k\cdot(-\Gamma_{\Phi,d}\nabla\Phi) \quad dS\\ &+\sum_{k=\varphi,\phi}\int_{V_k(t)}S_\Phi(\Phi) \quad dV\\ &+\frac{1}{2}\sum_{k=\varphi,\phi}\sum_{j=\phi,\varphi}(1-\delta_{j,k})\int_{S_{I,jk}(t)}S_{\Phi,I}(\Phi)dS\end{aligned} \tag{2.7}$$

The first three terms hold the already introduced bulk (phase interior) contributions within the phases $\varphi$ and $\phi$, respectively. The last term on the r.h.s. of equation (2.7) represents the coupling interfacial source term. $\delta_{j,k}$ herein denotes Kronecker's delta. Analogously to the procedure followed in the previous section the Leibniz' Theorem (Eq. (2.2)) is applied to equation (2.7).

$$\begin{aligned}\frac{d}{dt}\int_{V_\varphi(t)}\rho\Phi \quad dV = &\int_{V_\varphi(t)}\frac{\partial}{\partial t}(\rho\Phi) \quad dV\\ &+\int_{S_{I,\varphi\phi}(t)}\vec{n}_{I,\varphi\phi}\cdot\vec{U}_{I,\varphi\phi}(\rho\Phi) \quad dS\end{aligned} \tag{2.8}$$

Where the second term cannot be neglected any more since the interface velocity is not null ($\vec{U}_{I,\varphi\phi} \neq 0$). Applying Gauss' theorem as well (Eqns.

(2.3) and (2.4)) lead to the simplified version of the two-phase generalized transport equation:

$$\sum_{k=\varphi,\phi}\int_{V_k(t)}\left[\frac{\partial}{\partial t}(\rho\Phi)+\nabla\cdot(\rho\Phi\vec{U})-\nabla\cdot(\Gamma_{\Phi,d}\nabla\Phi)-S_\Phi(\Phi)\right]\quad dV=$$
$$\frac{1}{2}\sum_{k=\varphi,\phi}\sum_{j=\phi,\varphi}(1-\delta_{j,k})\int_{S_{I,jk}(t)}\Big[\rho\Phi(\vec{U}-\vec{U}_{I,kj})\cdot\vec{n}_{I,kj} \tag{2.9}$$
$$-(\Gamma_{\Phi,d}\nabla\Phi)\cdot\vec{n}_{I,kj}+S_{\Phi,I}(\Phi)\Big]\quad dS$$

As equation (2.9) is valid for arbitrary phase volumes $V_k$ and arbitrary configurations of phase interfaces $S_{I,\varphi\phi}$ within $V$, the two terms and therein the integrands can be set to zero interchangeably. Thus, the local instantaneous generic transport equation can be deduced from the first integrand (representing the bulk phase contributions). As it has already been the result for the single phase case (equation (2.6)). However, considering the two or multi-phase case, there is always coupling with (at least) another transport equation of the same type due to the presence of a neighbouring phase $\phi$. Hence, interfacial conditions have to be specified from the r.h.s. term of equation (2.9) in order to close the system.

# 3 Governing Equations

## 3.1 Averaged Continuity Equation

In this section the averaged continuity equation for a compressible two-fluid system is described. The formulation is based on the conditional averaging of the continuity equation based on the work of Weller [14] as described in [15].

$$\frac{\partial\rho_\varphi\alpha_\varphi}{\partial t}+\nabla\cdot(\rho_\varphi\vec{U}_\varphi\alpha_\varphi)=\Gamma_\varphi \tag{2.10}$$

where $\varphi$ identifies the phase, $\alpha$ is the phase fraction and $\Gamma$ represents the mass source term.

It is possible to write a phase-continuity equation for each of the two phases ($\varphi\in[1,2]$) and combine them to obtain the volumetric continuity equation for the mixture[1]:

$$\frac{\partial(\rho_1\alpha_1)}{\partial t}+\nabla\cdot\left(\rho_1\alpha_1\vec{U}_1\right)=0 \tag{2.11}$$
$$\frac{\partial(\rho_2\alpha_2)}{\partial t}+\nabla\cdot\left(\rho_2\alpha_2\vec{U}_2\right)=0 \tag{2.12}$$

[1]In this case the source term $\Gamma_\varphi$ was neglected, hence the formulation of the two-fluid model is for the non reacting system

The phase fractions are calculated through the solution of one of these phase continuity equations. Its discretization must bring a conservative and bounded solution.
The continuity equation for phase 1 can be re-written as:

$$\rho_1 \frac{\partial \alpha_1}{\partial t} + \alpha_1 \frac{\partial \rho_1}{\partial t} + \rho_1 \nabla \cdot (\alpha_1 \vec{U}_1) + \alpha_1 \vec{U}_1 \cdot \nabla(\rho_1) = 0 \tag{2.13}$$

Introducing the total derivative for the density and dividing both terms by the density itself we obtain:

$$\frac{\partial \alpha_1}{\partial t} + \nabla \cdot (\alpha_1 \vec{U}_1) + \frac{\alpha_1}{\rho_1} \frac{D(\rho_1)}{Dt} = 0 \tag{2.14}$$

Weller [14] proposed a different formulation of the phase continuity equation, so that all terms are conservative and $\alpha_1$ is bounded.

$$\frac{\partial (\alpha_1)}{\partial t} + \nabla \cdot \left(\alpha_1 \vec{U}\right) + \nabla \cdot \left(\alpha_1 \vec{U}_r (1 - \alpha_1)\right) + \frac{\alpha_1}{\rho_1} \frac{D(\rho_1)}{Dt} = 0 \tag{2.15}$$

Where:

$$\vec{U} = \alpha_1 \vec{U}_1 + \alpha_2 \vec{U}_2 \tag{2.16}$$

$$\vec{U}_r = \vec{U}_1 - \vec{U}_2 \tag{2.17}$$

This rearranged equation couples implicitly the two phases through the introduction of these two velocities.
The same substitution can be made for phase 2:

$$\frac{\partial (\alpha_2)}{\partial t} + \nabla \cdot \left(\alpha_2 \vec{U}\right) - \nabla \cdot \left(\alpha_1 \vec{U}_r (1 - \alpha_1)\right) + \frac{\alpha_2}{\rho_1} \frac{D(\rho_2)}{Dt} = 0 \tag{2.18}$$

Since $\vec{U}_2 = \vec{U} - \alpha_1 \vec{U}_r$.
Equations (2.15) and (2.18) are summed to obtain:

$$\nabla \cdot \vec{U} = -\frac{\alpha_1}{\rho_1} \frac{D(\rho_1)}{Dt} - \frac{\alpha_2}{\rho_2} \frac{D(\rho_2)}{Dt} \tag{2.19}$$

Substituting equation (2.19) in (2.15) and rearranging the equation as follows, we obtain:

$$\nabla \cdot (\alpha_\varphi \vec{U}) = \alpha_\varphi \nabla \cdot (\vec{U}) + \vec{U} \cdot \nabla \alpha_\varphi \tag{2.20}$$

$$\begin{aligned} \frac{\partial (\alpha_1)}{\partial t} + \vec{U} \cdot \nabla \alpha_1 + \nabla \cdot \left(\alpha_1 \vec{U}_r (1 - \alpha_1)\right) = \\ \alpha_1 \alpha_2 \left( \frac{1}{\rho_2} \frac{D(\rho_2)}{Dt} - \frac{1}{\rho_1} \frac{D(\rho_1)}{Dt} \right) \end{aligned} \tag{2.21}$$

Re-writing (2.21) through (2.20) we obtain the final equation:

$$\frac{\partial(\alpha_1)}{\partial t} + \nabla \cdot (\alpha_1 \vec{\bar{U}}) + \nabla \cdot \left(\alpha_1 \vec{\bar{U}}_r (1-\alpha_1)\right) - \alpha_1 \nabla \cdot (\vec{\bar{U}}) = \alpha_1 \alpha_2 \left( \frac{1}{\rho_2} \frac{D(\rho_2)}{Dt} - \frac{1}{\rho_1} \frac{D(\rho_1)}{Dt} \right) \tag{2.22}$$

## 3.2 Averaged Momentum Equation

Following the analysis carried on in the previous section 3.1, conditional averaging is performed also on the momentum equation derived from the generalised transport equation seen in section 2:

$$\frac{\partial(\alpha_\varphi \rho_\varphi \vec{\bar{U}}_\varphi)}{\partial t} + \nabla \cdot (\alpha_\varphi \rho_\varphi \vec{\bar{U}}_\varphi \vec{\bar{U}}_\varphi) + \nabla \cdot (\alpha_\varphi \bar{\mathcal{R}}_\varphi^{eff}) = -\alpha_\varphi \nabla \bar{p} + \alpha_\varphi \rho_\varphi \vec{g} + \bar{\mathcal{M}}_\varphi \tag{2.23}$$

where $\bar{\mathcal{R}}_\varphi^{eff}$ is the combined Reynolds and viscous stress tensor [16, 17] and $\bar{\mathcal{M}}_\varphi$ represents the averaged inter-phase momentum transfer term for both dispersed and segregated flows.
This last term has to be defined, and hence modelled for each phase. In order to obtain a conservative solution the total momentum transferred through phases has to be null. This means that the summation of each momentum transfer contribution for both phases has to be zero:

$$\sum_{\substack{\varphi \in [1,2] \\ i \in [IMTM]}} \bar{\mathcal{M}}_{\varphi_i}{}^2 = 0 \tag{2.24}$$

Since the system is a two-fluid one, it is sufficient to define one expression for the inter-phase momentum transfer to reach closure of the system. Usually the models are fitted to describe the forces acting on the dispersed phase, or better to the Dispersed Phase Elements.
It is important to notice that during the simulations, in different regions, the dispersed phase isn't always the same and this is why the models has to be defined for both phases and tuned to the case study each time.

## 3.3 Averaged Energy Equation

The simulations considered in this work deals mainly with isothermal flows, the possible temperature fluctuations are given mainly by expansion and

[2]IMTM: interfacial momentum transfer models

phase change (i.e. evaporation and condensation) phenomena. Even though the flow is roughly isothermal an energy equation is needed to close the system, because both phases in Eulerian-Eulerian approach are compressible. The two enthalpy equations are defined as:

$$\begin{aligned}\frac{\partial(\alpha_\varphi\rho_\varphi e_\varphi)}{\partial t}+\nabla\cdot(\alpha_\varphi\rho_\varphi e_\varphi\vec{\bar{U}}_\varphi)+\frac{\partial\left(\alpha_\varphi\rho_\varphi K_\varphi\right)}{\partial t}+\nabla\cdot\left(\alpha_\varphi\rho_\varphi K_\varphi\vec{\bar{U}}_\varphi\right)=\\ -\nabla\cdot\left[\alpha_\varphi\left(q_\varphi+q_\varphi^t\right)\right]+\nabla\cdot\left(\alpha_\varphi T_\varphi\vec{\bar{U}}_\varphi\right)+\alpha_\varphi\rho_\varphi\cdot\vec{\bar{U}}_\varphi+S_\varphi\end{aligned} \tag{2.25}$$

Where $e_\varphi$ consists in the standard thermal energy and the turbulent kinetic energy, $K_\varphi$ is the kinetic energy of phase $\varphi$ given by $K_\varphi = 0.5\cdot\vec{U}_\varphi^2$, $S_\varphi$ represents the interfacial supply energy to $\varphi^{th}$ phase and $q_\varphi^t$ takes account for the turbulent energy convection as well as for the turbulent work. In solving problems, it is often useful to separate the mechanical and thermal effects in the total energy equation. Thus from the standard method of dotting the momentum equation by the velocity, we have the mechanical energy equation [21]:

$$\begin{aligned}\frac{\partial\left(\alpha_\varphi\rho_\varphi\frac{\vec{\bar{U}}_\varphi^2}{2}\right)}{\partial t}+\nabla\cdot\left(\alpha_\varphi\rho_\varphi\frac{\vec{\bar{U}}_\varphi^2}{2}\vec{\bar{U}}_\varphi\right)=&-\vec{\bar{U}}_\varphi\cdot\nabla(\alpha_\varphi p_\varphi)+\vec{\bar{U}}_\varphi\cdot\nabla\cdot\left(\alpha_\varphi\mathcal{R}_\varphi^{eff}\right)\\ &+\alpha_\varphi\rho_\varphi g\cdot\vec{\bar{U}}_\varphi+\bar{\mathcal{M}}_\varphi\cdot\vec{\bar{U}}_\varphi\end{aligned} \tag{2.26}$$

Then by subtracting equation (2.26) to (2.25) the internal energy equation can be obtained:

$$\begin{aligned}\frac{\partial(\alpha_\varphi\rho_\varphi e_\varphi)}{\partial t}+\nabla\cdot(\alpha_\varphi\rho_\varphi e_\varphi\vec{\bar{U}}_\varphi)=&-\nabla\cdot(\alpha_\varphi p_\varphi)\\ -\nabla\cdot\left[\alpha_\varphi(q_\varphi^t+\mathcal{R}_\varphi^t\cdot\vec{\bar{U}}_\varphi)\right]&-\alpha_\varphi p_\varphi\nabla\cdot\vec{\bar{U}}_\varphi\\ +\alpha_\varphi\left(\mathcal{R}_\varphi^{eff}\right)\times\nabla\vec{\bar{U}}_\varphi&+(S_\varphi-\bar{\mathcal{M}}_\varphi\cdot\vec{\bar{U}}_\varphi)\end{aligned} \tag{2.27}$$

For two-phase flow analyses, the enthalpy energy equation is important and it is frequently used to solve various engineering problems. Thus the virtual enthalpy $h_\varphi$ is defined

$$h_\varphi = i_\varphi+\frac{\vec{\bar{U}}_\varphi^2}{2}=e_\varphi+\frac{p_\varphi}{\rho_\varphi} \tag{2.28}$$

substituting (2.28) in (2.27) we obtain:

$$\begin{aligned}\frac{\partial(\alpha_\varphi \rho_\varphi h_\varphi)}{\partial t} + \nabla \cdot (\alpha_\varphi \rho_\varphi h_\varphi \vec{U}_\varphi) &= -\nabla \cdot (\alpha_\varphi p_\varphi) \\ -\nabla \cdot \left[\alpha_\varphi (q^t_\varphi + \mathcal{R}^t_\varphi \cdot \vec{U}_\varphi)\right] &+ \frac{D_\varphi}{Dt}(\alpha_\varphi p_\varphi) \\ +\alpha_\varphi \left(\mathcal{R}^{eff}_\varphi\right) \times \nabla \vec{U}_\varphi &+ (S_\varphi - \mathcal{M}_\varphi \cdot \vec{U}_\varphi)\end{aligned} \tag{2.29}$$

These thermal energy equations are extremely complicated due to the interactions between the mechanical terms from the turbulent fluctuations and the thermal terms. However, in many practical two-phase flow problems, the heat transfer and the phase change terms dominate the energy equations. In such a case, the above equations can be reduced to simpler forms.

## 3.4 Turbulence Modeling

Turbulence modelling is needed in order to determine the Reynolds' stresses tensor in the momentum equation and in particular the kinematic viscosity. This work will focus solely on the formulation of the $k-\epsilon$ turbulence model for two-phase system, but it is important to highlight that also $k-\omega$ and LES models are available in OpenFOAM [34]. The $k-\epsilon$ model solves two additional transport equations for [16]:

$k$ : Kinetic Energy;

$\epsilon$ : Turbulent Dissipation.

From these two quantities the kinematic turbulent viscosity can be calculated as follows:

$$\nu^t = C_\mu \frac{k^2}{\epsilon} \tag{2.30}$$

Where $C_\mu = 0.09$. Two $k-\epsilon$ models will be presented in the following paragraphs, for other models and a thorough description of turbulence in multiphase flows it is possible to consult [2, 15].

### 3.4.1 Standard $k-\epsilon$ with RDT-based compression term

This model aims to extend the validity of the standard $k-\epsilon$ model also to compressible flows through the model proposed by Sherif H. El Tahry [35]. With this approach the transport equations for $k$ and $\epsilon$ are similar to the

ones derived in [16,17] for single phase fluids but present, as well, additional terms that accounts for the fluid compressibility:

$$\frac{\partial(\alpha_\varphi \rho_\varphi k_\varphi)}{\partial t} + \nabla \cdot (\alpha_\varphi \rho_\varphi k_\varphi \vec{\bar{U}}_\varphi) - \nabla \cdot (\alpha_\varphi \rho_\varphi D_k^{eff} k_\varphi) = \alpha_\varphi \rho_\varphi \mathcal{G} \underbrace{-\frac{2}{3}\alpha_\varphi \rho_\varphi \nabla \vec{\bar{U}}_\varphi \cdot k_\varphi}_{\text{Compressibility Term}} - \alpha_\varphi \rho_\varphi \epsilon_\varphi \tag{2.31}$$

$$\frac{\partial(\alpha_\varphi \rho_\varphi \epsilon_\varphi)}{\partial t} + \nabla \cdot (\alpha_\varphi \rho_\varphi \epsilon_\varphi \vec{\bar{U}}_\varphi) - \nabla \cdot (\alpha_\varphi \rho_\varphi D_\epsilon^{eff} \epsilon_\varphi) = C_1 \alpha_\varphi \rho_\varphi \mathcal{G} \frac{\epsilon_\varphi}{k_\varphi} \underbrace{-\left(\frac{2}{3}C_1 - C_3\right)\alpha_\varphi \rho_\varphi \nabla \vec{\bar{U}}_\varphi \cdot \epsilon_\varphi}_{\text{Compressibility Term}} - C_2 \alpha_\varphi \rho_\varphi \epsilon_\varphi \frac{\epsilon_\varphi}{k_\varphi} \tag{2.32}$$

Where:

$D_k^{eff}$ : Effective diffusivity for $k = \frac{\nu^t}{\sigma_k} + \nu$;

$D_\epsilon^{eff}$ : Effective diffusivity for $\epsilon = \frac{\nu^t}{\sigma_\epsilon} + \nu$;

$\sigma_k$ = 1;

$\sigma_\epsilon$ = 1.3;

$C_1$ = 1.44 ;

$C_2$ = 1.92;

$C_3$ = -0.33.

$\mathcal{G}$ stands for the production of turbulent kinetic energy and is defined as [2]:

$$\mathcal{G} = \nu^t \left[\nabla \vec{\bar{U}}_\varphi : dev(\nabla \vec{\bar{U}}_\varphi + (\nabla \vec{\bar{U}}_\varphi)^T)\right] \tag{2.33}$$

Where ":" stands for the double inner product and "dev" operator extracts the deviatoric component of a tensor:

$$dev(T) = T - \frac{1}{3} tr(T) I \tag{2.34}$$

It can be easily demonstrated [36] that equation (2.33) is derived from the substitution of Boussinesq assumption into the tensorial notation of the turbulent kinetic energy production term.

### 3.4.2 $k-\epsilon$ with RDT-based compression term and bubble generated turbulence

As illustrated by Lahey in [37] a source term accounting for the bubble generated turbulence is added in both transport equations described in section (3.4.1). The two terms $\Phi_k$ and $\Phi_\epsilon$ are defined so that:

$$\Phi_k = \frac{k_c}{C_{\epsilon_2}\epsilon_c}\Phi_\epsilon \tag{2.35}$$

$$\Phi_k = C_p\Big(1 + C_D^{4/3}\Big)\alpha_d\frac{|\vec{\bar{U}}_r^3|}{D_d} \tag{2.36}$$

Where $C_p = 0.25$ for potential flow around a sphere, $C_D$ is the drag coefficient, $U_r$ is the slip velocity between phases, the subscripts $c$ and $d$ refers respectively to continuous and dispersed phase and finally, $D_d$ refers to the Sauter diameter of the dispersed phase DPE.

# 4 Two-Phase Cavitating Systems

In this section will be reported the characteristic equations for a two-phase system in which cavitation occurs. In particular will be highlighted the various source terms that has to be added to the transport equations described in section 3.

## 4.1 Continuity Equation

For which concerns the continuity equation,it is exactly equation (2.10):

$$\frac{\partial\rho_\varphi\alpha_\varphi}{\partial t} + \nabla\cdot(\rho_\varphi\vec{\bar{U}}_\varphi\alpha_\varphi) = \Gamma_\varphi \tag{2.37}$$

Where the term $\Gamma_\varphi$ represents the mass souce term per unit time of phase $\varphi$.

## 4.2 Momentum Equation

The mass source term highlighted in section 4.1 has to be added to the formulation of the momentum equation in order to take into account the phase exchange:

$$\begin{aligned}\frac{\partial(\alpha_\varphi\rho_\varphi\vec{\bar{U}}_\varphi)}{\partial t} + \nabla\cdot(\alpha_\varphi\rho_\varphi\vec{\bar{U}}_\varphi\vec{\bar{U}}_\varphi) + \nabla\cdot(\alpha_\varphi\bar{\mathcal{R}}_\varphi^{eff}) = &-\alpha_\varphi\nabla\bar{p} + \alpha_\varphi\rho_\varphi\vec{g} \\ &+ \bar{\mathcal{M}}_\varphi + \Gamma_\varphi\vec{\bar{U}}_\varphi\end{aligned} \tag{2.38}$$

### 4.3 Energy Equation

The energy equation source term is a little bit more complicated from a computational point of view than the single phase solution [16, 17], since it has to take into account the latent heat of evaporation/condensation of the cavitating fluid. The actual analytical formulation doesn't change significantly with rispect to (2.25):

$$\begin{aligned}\frac{\partial(\alpha_\varphi\rho_\varphi e_\varphi)}{\partial t}+\nabla\cdot(\alpha_\varphi\rho_\varphi e_\varphi\vec{U}_\varphi)+\frac{\partial\left(\alpha_\varphi\rho_\varphi K_\varphi\right)}{\partial t}+\nabla\cdot\left(\alpha_\varphi\rho_\varphi K_\varphi\vec{U}_\varphi\right)=\\ -\nabla\cdot\left[\alpha_\varphi\left(q_\varphi+q_\varphi^t\right)\right]+\nabla\cdot\left(\alpha_\varphi T_\varphi\vec{U}_\varphi\right)+\alpha_\varphi\rho_\varphi\cdot\vec{U}_\varphi+S_\varphi+L_\varphi\Gamma_\varphi\end{aligned} \tag{2.39}$$

where, as said before, $L_\varphi$ represents the latent heat of evaportation/condensation.

# 5 Interfacial Momentum Transfer Modelling

As was introduced before in the chapter (section 3.2) it is important to describe the various interface momentum transfer terms. Before introducing all this different contributions it is also worth to highlight the substantial mathematical and physical differences between segregated and dispersed flows.
For a purpose of a consistent and general closure, the use of a scale similarity hypothesis is used, analogously to what is done in LES modeling for single-phase flows:

*The smallest resolved scales of a two-phase flow are assumed to exhibit similar characteristics as the largest non resolved ones.*

Following this hypothesis, it is reasonable and advantageous to consider dispersed flows within an *interfacial scale averaging closure framework* and segregated flows within an *interfacial resolving closure framework* [19].

In this section the two diverse framework will be described and the deriving inter-phase change terms will be listed. For a thorough analysis and derivation of the characteristic transport equation, the reader is suggested to refer to the work of Marschall [19]. Starting from the momentum averaged transport equation as derived from Marschall (see Eq. (2.38)) the source term will be described for both dispersed and segregated flows as the summation of two components namely the net force contribution from interfacial averaged pressure in case there is a gradient in the volumetric phase fraction

and the contribution due to shear stress and unbalanced interfacial pressure.

$$\bar{\mathcal{M}}_{\varphi} = \bar{\mathcal{M}}_{\sigma,\varphi} + \bar{\mathcal{M}}_{h,\varphi} \tag{2.40}$$

The first source term takes into account phase slip and surface tension effects,while the second term for the dispersed two-fluid system represents the sum of fluid dynamic forces exerted to a DPE and for the segregated flow type it just models the interfacial force density due to unbalanced pressures and viscous stresses, which manifest itself in a dissipative drag due to interfacial friction in the presence of phase slip.

## 5.1 Dispersed Flows

We will now focus on the dispersed Two-fluid model and a brief description of both source terms will be presented. Closure models for dispersed flows are well-established and widely used. Modelling is commonly accomplished employing two-fluid model frameworks, that are based on the concept of interpenetrating continua: The actual interface morphology and motion is not explicitly resolved, i.e., the averaging volume is chosen much larger than the characteristic length scale of the dispersed phase (e.g., the equivalent diameter of the (fluid) particle). As a consequence, all interactions between the phases have to be modelled appropriately within the averaging framework as is graphically described in figure 2.6. For which concerns the surface tension

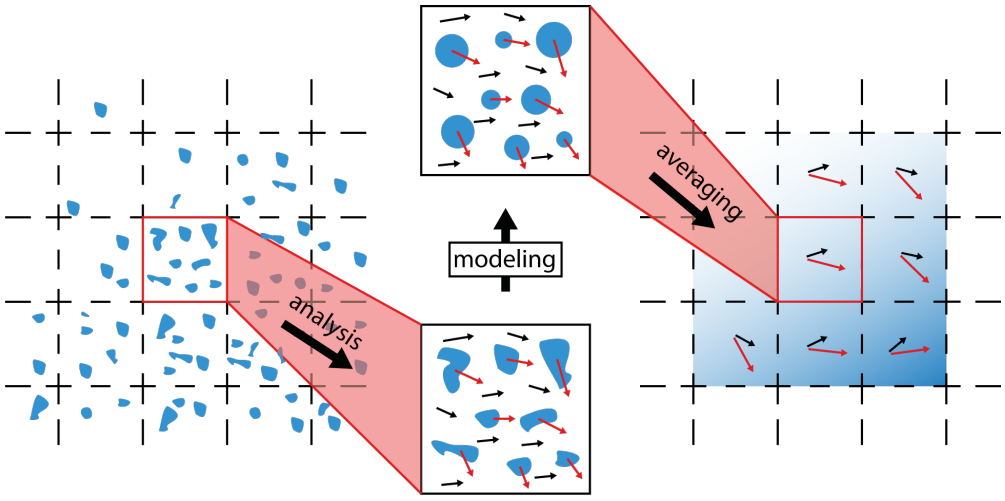


Figure 2.6: Graphical representation of interfacial scale averaging closure framework for dispersed two-fluid systems

term since the size of the averaging volume has been chosen significantly

larger than the characteristic length scale of the dispersed phase (e.g., the equivalent diameter of the fluid particles) it is justified to neglect fluid particles that intersect the control volume's surface, stating that there is no net force that has to be explicitly taken into account.

On the other hand the main contributions to inter-phase momentum transfer are due to the drag and lift force between the two phases, virtual mass forces, wall lubrication forces and turbulent dispersion (derived from the averaging process). As it is possible to imagine, there are several ways to take into account for interfacial momentum transfer [14,15,38] but, as stated before this work will focus on the formulation implemented in OpenFOAM and in particular on the formulation of Gosman et al. [39].
In any case it is interesting to write the inter-phase momentum transfer term in a general form and then, highlight singularly each characteristic constant through the diverse models available in literature. It reads:

$$\begin{aligned}
\bar{\mathcal{M}}_1 &= A_d\vec{\bar{U}}_r && \text{Drag} \\
&+A_{vm}\left(\frac{D_2\vec{U}_2}{Dt}-\frac{D_1\vec{U}_1}{Dt}\right) && \text{Virtual Mass} \\
&+A_{l_2}\vec{\bar{U}}_r\times(\nabla\times\vec{\bar{U}}_2) && \text{Lift} \\
&+A_{l_1}\vec{\bar{U}}_r\times(\nabla\times\vec{\bar{U}}_1) && \\
&+A_\alpha\nabla\alpha && \text{Turbulent Dispersion} \\
&+\mathcal{M}_1^{wl} && \text{Wall Lubrication}
\end{aligned} \tag{2.41}$$

Neglecting the wall lubrication for a first analysis, the $A_i$ coefficients are modelled differently in literature [14, 15, 38], following in table 2.2 are presented the coefficients derived in [39] by Gosman.

| Coefficient | $A_d$ | $A_{l_1}$ | $A_{l_2}$ | $A_{vm}$ | $A_\alpha$ |
|---|---|---|---|---|---|
| Gosman Model | $\alpha\frac{3}{4}\frac{\rho_2}{d_1}C_d\lvert\vec{\bar{U}}_r\rvert$ | 0 | $\alpha_1 C_l\rho_2$ | $\alpha_1 C_{vm}\rho_2$ | $\frac{3}{4}C_d\frac{\rho_2}{d}\frac{\nu^t}{\sigma_\alpha}\lvert\vec{\bar{U}}_r\rvert$ |

Table 2.2: Multiplier terms for Gosman et al. [39]

In the following paragraphs a detailed formulation of the most important interface momentum transfer coefficients models will be carried on, and their schematic representation is presented in figure (2.7).

## 5.2 Lift Force

For multiphase flows, it is recommended to include the effect of lift forces on the dispersed phase particles, droplets, or bubbles. These lift forces act

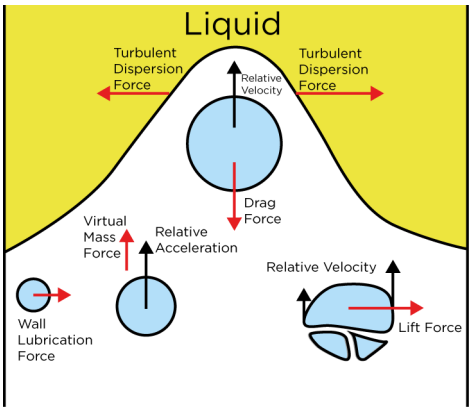


Figure 2.7: Graphical representation of most important momentum transfer phenomena.

on a particle mainly due to velocity gradients in the continuous phase flow field as can be seen in figure (2.8). The lift force will be more significant for larger particles, but the model assumes that the particle diameter is much smaller than the inter particle spacing. Thus, the inclusion of lift forces is not appropriate for closely packed particles or for very small ones [20].

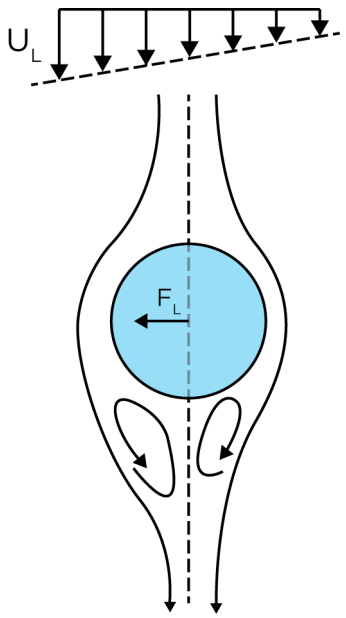


Figure 2.8: Lift force generated on a bubble due to the linear shear flow of the fluid around it.

### 5.2.1 The Lift Force Formulation

the lift force acting on the dispersed phase $d$ in the continuous phase $c$ can be calculated as:

$$\vec{F}_{lift} = -C_l \rho_c \alpha_d (\vec{v}_c - \vec{v}_d) \times (\nabla \times \vec{v}_c) \tag{2.42}$$

Where:

$C_l$= Lift coefficient that has to be modelled (sec 1);

$\rho_c$= Continuous phase density;

$\alpha_d$= Dispersed phase volume fraction;

$\vec{v}_c$= Continuous phase velocity;

$\vec{v}_d$= Dispersed phase velocity.

This factor will be added on the right-hand side of the momentum equation for each phase, remembering that $\vec{F}_{lift,c} = -\vec{F}_{lift,d}$ as stated before in section 3.2.

In most cases, the lift force is insignificant compared to the drag force, so there is no reason to include this extra term. If the lift force is significant (for example, if the phases separate quickly), it may be appropriate to include this term. In This study the Tomiyama model was considered [40] and the reasons of this choice will be highlighted in the following paragraph. An overview of other lift models is presented in appendix B.

**5.2.1.1 Tomiyama Model** Tomiyama derived through a series of experiments on a water-glycerol mixture a lift coefficient for different Reynolds (low Reynolds regime) and Eötvos regimes. It has been clarified through a number of experiments that the lateral migration strongly depends on bubble size and shape, i.e., small bubbles tend to migrate toward the pipe wall which causes a wall-peak bubble distribution, whereas large bubbles tend to migrate toward the pipe center which results in a core-peak bubble distribution. These experiments lead to the formulation of the lift coefficient as [40]:

$$C_l = \begin{cases} min(0.288 \cdot tanh(0.121 Re_b), f(Eo_d)) & Eo_d < 4 \\ f(Eo_d) & 4 < Eo_d < 10 \\ -0.27 & Eo_d > 10 \end{cases} \tag{2.43}$$

Where $Re_b$ is the bubble Reynolds number, and $Eo_d$ is the Eötvos number based on Wellek horizontal bubble diameter $d_h$:

$$Eo_d = \frac{(\rho_2 - \rho_1) g d_h^2}{\sigma} \tag{2.44}$$

$$d_h = D_S(1 + 0.163 Eo^{0.757})^{1/3} \tag{2.45}$$

With this formulation the lift coefficient changes sign at $D_S > 5.8mm$ as showed in figure (2.10) and (2.9)so that large bubbles moved towards the centerline of the pipe as Tomiyama experimented.

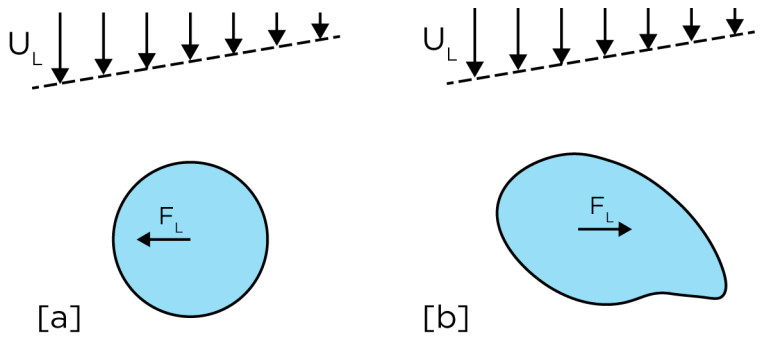


Figure 2.9: Lift force on undistorted (a) and distorted (b) bubble in a shear flow

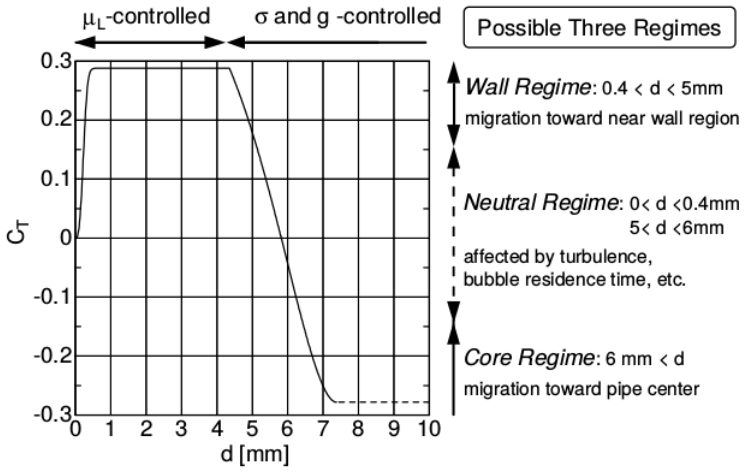


Figure 2.10: Lift coefficient regimes as described by Tomiyama [40]

## 5.3 Wall Lubrication Force

For liquid-gas bubbly flows using the Eulerian model it is recommended to include the effect of wall lubrication forces on the dispersed phase (bubbles). The wall lubrication force tends to push the bubbles away from walls. For example, in a bubbly upward flow in a vertical pipe this force results in the dispersed phase concentrating in a region near, but not immediately adjacent to, the wall as in figure (2.11).

### 5.3.1 The Wall Lubrication Force Formulation

The Wall lubrication force acting on a dispersed phase $d$, in a continuous phase $c$, has the following form:

$$\vec{F}_{wl} = C_{wl}\rho_c\alpha_d|(\vec{v}_c - \vec{v}_d)_{||}|^2\vec{n}_w \tag{2.46}$$

Where:

$C_{wl}$: Wall lubrication coefficient that has to be modeled(sec 2);

$\rho_c$: Continuous phase density;

$\alpha_d$: Dispersed phase volume fraction;

$|(\vec{v}_c - \vec{v}_d)_{||}|$: Phase relative velocity component tangential to the wall surface;

$\vec{n}_w$: Unitary vector pointing away from the wall.

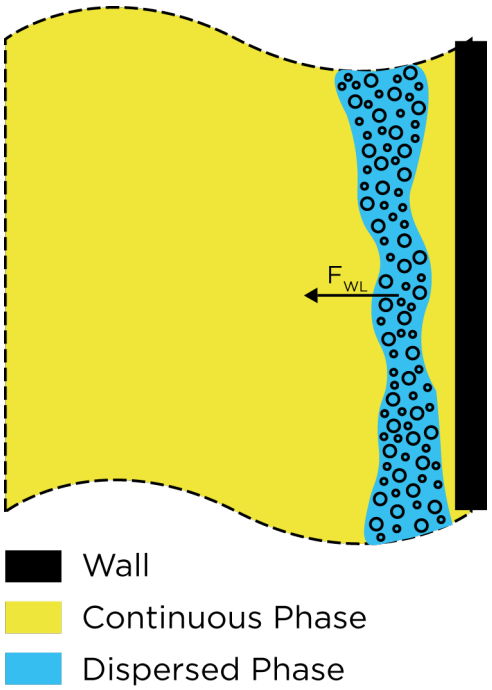


Figure 2.11: Continuous phase lubrication layer between the wall and the dispersed phase, i.e. the gaseous phase

Since this contribution is more influencing in cases with gas and/or liquid wall adherence is present, the effect of wall lubrication will be neglected in the simulations presented in chapter 4. If the reader is interested in wall lubrication modelling it is suggested to refer to appendix B.

## 5.4 Drag Force

This force represents the resistance opposed to the bubble motion in the fluid (or, more generally, the resistance of the relative motion between two phases). The drag force clearly depends on the bubble's size (i.e. a larger bubble experiences a larger drag force) and on the relative velocity between the two phases $\vec{v}_r = \vec{v}_1 - \vec{v}_2$.

### 5.4.1 Drag Formulation

For fluid-fluid flows, each dispersed phase is assumed to form droplets or bubbles. This has an impact on how each of the fluids is assigned to a

particular phase. For example, in flows where there are unequal amounts of two fluids, the predominant fluid should be modelled as the continuous fluid, since the sparser fluid is more likely to form droplets or bubbles. The Drag coefficient for these types of mixtures can be written in the following general form:

$$\vec{F}_{drag} = -\frac{3}{4}\frac{C_d}{D_s}\rho_c\alpha_d|\vec{v}_r|\vec{v}_r \tag{2.47}$$

Where:

$D_s$: is the mean Sauter diameter;

$C_d$: is the drag coefficient, described differently for the different models;

$\rho_c$: is the density of the continuous phase;

$\alpha_d$: is the void fraction of the dispersed phase;

### 5.4.2 Drag Models

The drag coefficient is usually deduced from experiments and many models were developed in order to fit different experimental data-set. Since the derivation of the model is strictly correlated to the data-set used to derive it, the constant that is calculated from it results really case-dependent. This leads to a vast selection of models. It is crucial to select the right model when the case results highly drag-dependent because, otherwise the solution could result distorted. Following is presented the most used drag model for dispersed flows, namely the *Schiller Neumann* model [41], as it will be described in chapter 4 the liquid phase needs to be accounted as locally dispersed and hence a drag model needs to be defined for it. The other important drag model is, as implemented in OpenFOAM, the *saturated* that will be described thoroughly at the end of this chapter. If the reader is interested to deepen the topic can refer to appendix B, where the other drag models implemented in OpenFOAM are described.

**5.4.2.1 Schiller and Neumann Model** It is the most general method, and is used for all fluid-fluid pairs of phases. For this model the drag function is [41]:

$$C_d = \frac{24}{Re_b}(1 + 0.15Re_b^{0.687}) \tag{2.48}$$

Where the bubble Reynolds number is defined as $Re_b = \frac{|\vec{v}_1 - \vec{v}_2|D_S}{\nu_b}$. This formulation of the drag coefficient is valid only for solid spherical particles

with $Re_b < 1000$. For $Re_b > 1000$ the drag coefficient is nearly constant and equal to 0.44, therefore the following model is implemented:

$$C_d = max\left(\frac{24}{Re_b}(1 + 0.15Re_b^{0.687}), 0.44\right) \tag{2.49}$$

## 5.5 Turbulent Dispersion Force

For multiphase turbulent flows using the Eulerian model, it is possible to include the effects of turbulent dispersion forces which account for the interphase turbulent momentum transfer. These forces acts as a turbulent diffusion in dispersed flows. This terms, hence accounts for the turbulent fluctuations of liquid velocity and the effect that has on the gas bubbles. The characterizing feature of this force is the fact that it is proportional to the void-fraction gradient and therefore this could theoretically generate unstable results [15].

### 5.5.1 Turbulent Dispersion Force Formulation

The turbulent dispersion force arises from averaging the interphase drag term. For a dispersed phase ($d$) and a continuous phase ($c$) the turbulent drag is modelled as:

$$K_{dc}(\tilde{\vec{v}}_d - \tilde{\vec{v}}_c) = K_{dc}(\vec{v}_d - \vec{v}_c) - K_{dc}\vec{v}_{dr} \tag{2.50}$$

From this averaging process two terms are obtained, the first right-hand side term is the mean momentum exchange between the two phases (see section 5.4) and the second term is the turbulent dispersion force, that can be expressed as:

$$\vec{F}_{td,c} = -\vec{F}_{td,d} = -K_{dc}\vec{v}_r \tag{2.51}$$

Where

> $\vec{v}_r$: is the drift velocity and accounts for the dispersion of secondary phases due to transport by turbulent fluid motion;

For an overview on the available models please refer to appendix B, for what concerns the simulations of chapter 4 no turbulent dispersion was considered since the benchmark simulation is laminar.

## 5.6 Virtual Mass Force

For multiphase flows, it is possible to include the virtual mass effect that occurs when a secondary phase accelerates relatively to the primary phase.

The inertia of the primary-phase mass encountered by the accelerating particles (or droplets or bubbles) exerts a virtual mass force on the particles. The virtual mass effect is significant when the dispersed phase density is much smaller than the continuous phase density.

### 5.6.1 Virtual Mass Force Formulation

The virtual mass force is defined as:

$$\vec{F}_{vm} = -C_{vm}\alpha_p\rho_q\left(\frac{D_q\vec{U}_c}{Dt} - \frac{D_p\vec{U}_d}{Dt}\right) \tag{2.52}$$

Where:

$C_{vm}$: is the virtual mass coefficient (usually set at 0.5);

$\frac{D_i(\phi)}{Dt} = \frac{\partial(\phi)}{\partial t} + \vec{U}_\varphi \cdot \nabla\phi$: phase material time derivative.

The only other model implemented in OpenFOAM is the one derived by Lamb [42], and it is based on the Interfacial area. The formulation is the following:

$$C_{vm} = \frac{(1-E)^{1/2} - E \cdot acos(E)}{E \cdot acos(E) - E \cdot (1-E)^{1/2}} \tag{2.53}$$

Where $E$ represent the aspect ratio that will be described in section 5.8. These models highlighted so far are the actual interfacial momentum exchange models. It is also interesting to pay attention to other two quantities to model as well: bubble diameter (also referred to as mean Sauter diameter) and the aspect ratio (an index to define how deformed is the bubble $E = \frac{D_h}{D_w}$). These two quantities has to be derived because are needed by most of the models mentioned in this section.

## 5.7 Bubble Diameter Modelling

In fluid dynamics the Sauter mean diameter is defined as the diameter of a sphere that has the same volume to surface ratio as the particles of interest. Three different models for the SMD will be listed in this section.

### 5.7.1 Constant SMD

The simplest way to model the dispersed phase particle diameter is through this model. A constant value set by the user is choosen for the diameter, this value should not be set arbitrarily but, instead should derive from tested correlations or experiments and be consistent with the assumptions made by the user.

### 5.7.2 Isothermal SMD

As it is possible to infer the diameter of the "mean" bubble is estimated through an isothermal relationship. The model is designed so that inputting a reference diameter ($d_0$) and a reference pressure ($p_0$) it is possible to derive the value of the SMD as function of the changing pressure:

$$SMD = d_0 \left( \frac{p_0}{p} \right)^{1/3} \tag{2.54}$$

Even though it provides a more accurate evaluation of the diameter of the bubble, since it adds the dependance on the system pressure, this model still highly relies on the initial assumptions and hence on the reference values for the diameter.

### 5.7.3 Interfacial Area Concentration Transport Equation

The interfacial area concentration ($a$) corresponds to the area of the dispersed phase bubbles per unit volume and for spherical bubbles is:

$$a = \frac{6\alpha_\varphi}{D_s} \tag{2.55}$$

where $D_s$ is the Sauter mean diameter. The transport equation of $a$ is used in order to determine the mean sauter diameter, and is formulated as derived in [21, 43] by Hibiki and Ishii:

$$\frac{\partial(a)}{\partial t} + \nabla(a\vec{\bar{U}}_\varphi) = \frac{2}{3}\frac{a}{\alpha_\varphi}\left( \frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot (\alpha_\varphi \vec{\bar{U}}_\varphi) \right) + \Phi_{WEC} + \Phi_{RC} + \Phi_{TB} \tag{2.56}$$

The right hand side accounts respectively for the contribution of phase change and expansion due to pressure-density change, the wake entrainment coalescence source/sink term, the random coalescence source/sink term and the turbulent breakup source/sink term.

**5.7.3.1 Wake Entrainment Coalescence** This term models the coalescence due to the acceleration of the following bubble in the wake of the preceding bubble (fig 2.12) and can be written as:

$$\Phi_{WEC} = 12 \cdot \Psi C_{we} C_D^{1/3} |\vec{\bar{U}}_r| k \tag{2.57}$$

where $C_D$ is the drag coefficient of the bubble, $\Psi$ is a constant defined as $\Psi = \frac{1}{36\pi}$ and $k$ is the surface curvature, from which can be computed the interfacial area concentration ($a = \alpha \cdot k$). Finally $C_{we}$ is the model constant defined for different cases in table (2.3).

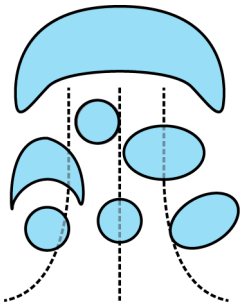

Figure 2.12: Scheme of Wake Coalescence phenomenon.

**5.7.3.2 Random Coalescence** This term models the coalescence through random collisions driven by turbulent eddies (fig 2.13) and can be written as:

$$\Phi_{RC} = 12 \cdot \frac{\Psi a C_{rc} |\vec{\bar{U}}_r|}{\alpha_{max}^{1/3}\left(\alpha_{max}^{1/3} - \alpha^{1/3}\right)} \left[1 - exp\left( - C \frac{\alpha_{max}^{1/3}\alpha^{1/3}}{\alpha_{max}^{1/3} - \alpha^{1/3}}\right)\right] \tag{2.58}$$

where $C_{rc}$ is the characteristic constant of the model, tabulated in tab (2.3) as well as $C$ and $\alpha_{max}$.

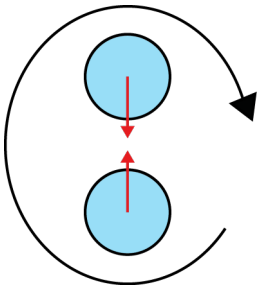

Figure 2.13: Scheme of Turbulent Eddy Coalescence phenomenon

**5.7.3.3 Turbulent Breakup** This term models the breakup due to the impact of turbulent eddies (fig 2.14) and can be written as:

$$\Phi_{TB} = 2 \cdot C_{ti} |\vec{\bar{U}}_r| \sqrt{1 - \frac{We_{cr}}{We}} exp\left( - \frac{We_{cr}}{We}\right) \qquad \text{if:} \quad We > We_{cr} \tag{2.59}$$

As said in the previous paragraphs the critical Weber number ($We_{cr}$) and $C_{ti}$ are tabulated in tab (2.3), whilst the bubble Weber number can be calculated as follows:

$$We = \frac{\rho v^2 l}{\sigma} \tag{2.60}$$

where:

| Mechanisms | Round Pipes vertical upward | Round Pipes vertical co-current downward | Confined Test Section vertical upward |
|---|---|---|---|
| Turbulent Breakup (Source) | 0.085 $We_{cr} = 6.0$ | 0.034 $We_{cr} = 6.0$ | 0.026 $We_{cr} = 8.0$ |
| Random Coalescence (Sink) | 0.004 $C = 3; \alpha_{max} = 0.75$ | 0.004 $C = 3; \alpha_{max} = 0.75$ | 0.003 $C = 3; \alpha_{max} = 0.75$ |
| Wake Entrainment Coalescence (Sink) | 0.002 | 0.002 | 0.042 |

Table 2.3: Summary of coefficients for source terms in interfacial area transport equation according to Ishii and Hibiki [43]

$\rho$: density of the phase considered;

v: phase velocity;

l: phase's characteristic length, usually bubble diameter (SMD);

$\sigma$: surface tension.

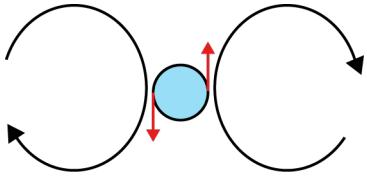

Figure 2.14: Scheme of Turbulent Breakup phenomenon

It can be tough as a measure of the relative importance of the fluid's inertia compared to its surface tension. It is useful in analysing multiphase flows with strongly curved surfaces. It is important to highlight that in the first term of the r.h.s is also accounted the source/sink term due to phase change.

## 5.8 Aspect Ratio Modelling

From a geometrical point of view the aspect ratio of a shape is the ratio of its sizes in different dimensions. In two-fluid modelling the aspect ratio is used to have informations about the shape of the bubble of the dispersed phase, in order to model more correctly the forces exchanged between phases. in fig 2.15 it is possible to see the difference between a spherical bubble (unitary aspect ratio) and an elliptical bubble (oriented horizontally). Several models are available in literature as well as experimental evidences. In this section some of those are explained.

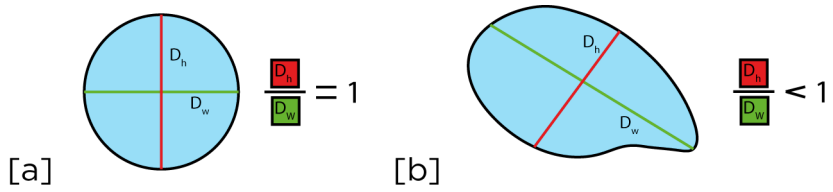


Figure 2.15: Aspect ratio for a spherical bubble (a) and for an elliptical bubble (b)

### 5.8.1 Constant Aspect Ratio

It is the simplest model with which is possible to deal with aspect ratio, it consist in a user defined value for the aspect ratio. Even tough it represents a strong simplification it is suggested for cases in which the shape of the bubble does not influence the interfacial momentum transfer and the bubbles can be considered spherical, i.e. $E = 1$.

### 5.8.2 Vakhrushev and Efremov Model

Vakhrushev and Efremov [44] derived the following formulation for the aspect ratio of a bubble of dispersed phase:

$$E = \begin{cases} 1 & Ta < 1 \\ \Big[0.81 + 0.206 \cdot tanh\Big[1.6 & \\ \quad -2 \cdot log_{10}\Big(max(Ta, 1)\Big)\Big]\Big] & 1 \leq Ta < 39.8 \\ 0.24 & 39.8 \leq Ta \end{cases} \tag{2.61}$$

Where $Ta$ represents the Takahashi number, defined as:

$$Ta = Re \cdot Mo^{0.23} \tag{2.62}$$

It is important to underline that this model is limited to low values of the Morton number in the order of $Mo \leq 10^{-3}$.

### 5.8.3 Wellek Model

Wellek [45] derived this formulation for the aspect ratio of a bubble of dispersed phase as a function of Eötvös number, $Eo$:

$$E = \frac{1}{1 + 0.163 \cdot Eo^{0.757}} \tag{2.63}$$

$$Eo = \frac{\Delta\rho g l^2}{\sigma} \tag{2.64}$$

### 5.8.4 Tomiyama Model

Tomiyama [46] derived this formulation for the aspect ratio of a bubble of dispersed phase as a function of the Sauter mean diameter of the bubble:

$$E = max\left(1 - 0.35 \cdot \frac{y_{Wall}}{D_s}, 0.65\right) \tag{2.65}$$

## 5.9 Segregated Flows

A closure for the second two-phase flow type under consideration (segregated flow) needs to be accomplished consistently, meaning also in the same manner as done for the dispersed two-phase flows. As a pivotal aspect of closure, a conceptual framework has to be formulated that is compatible and in the same spirit as the concept of interpenetrating continua for dispersed flow types. Such a conceptual framework is entitled as the concept of partially penetrating continua in the interfacial transition region: The interfacial morphology is partially resolved, i.e., the averaging volume is chosen sufficiently in its size to capture the main (mean) dynamics of the interfacial flow, whereas non-resolved interfacial morphologies and phase interactions again must be accounted for in the underlying averaging framework by appropriate (physical meaningful) closure models. As a consequence, the interface is represented as an interfacial transition region of certain characteristic width, which is determined by the averaging length scale as described in the following picture 2.16 [19]. As did before for the dispersed flow type, recalling

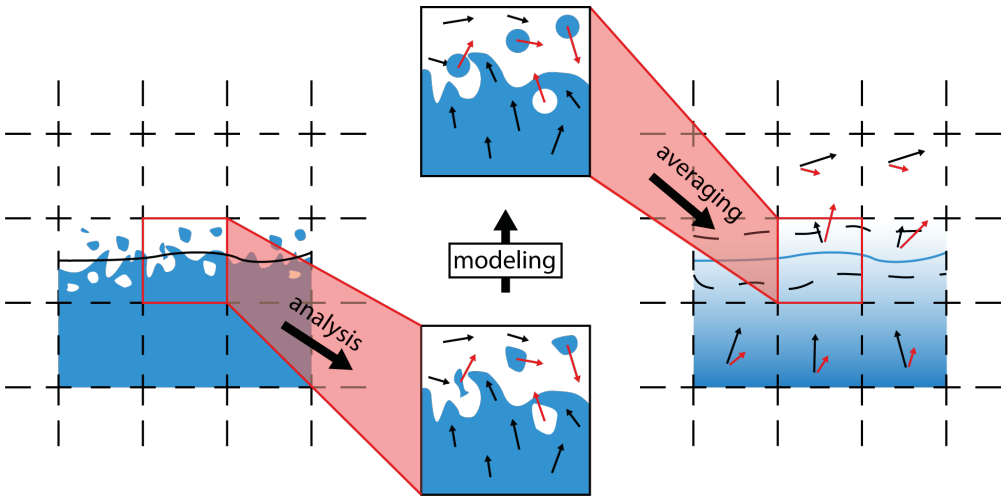


Figure 2.16: Graphical representation of interfacial resolving closure framework for segregated two-fluid systems

equation (2.40) in this paragraph the main source term for segregated flows are presented, for the complete derivation of this terms please refer to [19].

Since closure relations obviously are related to an interfacial force density due to surface tension, that is assumed to be dominated by non-resolved (microscopic/local) curvature, i.e., the inner morphological structure of the interfacial transition region, it is advisable to revisit the underlying assumption of

an isotropic interfacial topology: According to equation (2.66) the interfacial averaged curvature can be decomposed into a resolved mesoscopic contribution and a dominant non-resolved microscopic contribution $\frac{d\Sigma}{d\alpha_\varphi}$, where the interfacial area density $\Sigma$ is only allowed to vary across the interfacial transition region, that is in the direction of $\widehat{n_I}$, perpendicular to contours of constant $\alpha_\varphi$.

$$\widehat{\mathcal{K}_I} = \frac{\nabla^2 \alpha_\varphi}{|\nabla \alpha_\varphi|} + \frac{d\Sigma}{d\alpha_\varphi} \tag{2.66}$$

In the context of interfacial surface tension modelling it is proposed that:

$$\begin{aligned} \bar{\mathcal{M}}_{\varphi,\sigma} &= \alpha_\varphi \cdot 4\alpha_\varphi \alpha_\phi \bar{\mathcal{M}}_\sigma \\ \bar{\mathcal{M}}_{\phi,\sigma} &= \alpha_\phi \cdot 4\alpha_\phi \alpha_\varphi \bar{\mathcal{M}}_\sigma \end{aligned} \tag{2.67}$$

where $4\alpha_\varphi\alpha_\phi$ represents a regularization term. This is clearly stating that the essential contribution of the interfacial force density due to surface tension to the momentum equation of a particular phase is to be localized in a part of the interfacial transition region that does feature the inner curvature: on the adjacent sites of the interface (from the respective "phases view point"). It is just left to define $\bar{\mathcal{M}}_\sigma$:

$$\begin{aligned} \bar{\mathcal{M}}_\sigma &= \sigma \widehat{\mathcal{K}_I}\, \widehat{n_I}\, \Sigma + \sigma \frac{d\Sigma}{d\alpha_\varphi} \nabla \alpha_\varphi \\ &\approx \left( \nabla \cdot \left( \frac{\nabla \alpha_\varphi}{|\nabla \alpha_\varphi|} \right) + 4\Sigma_0 (1 - 2\alpha_\varphi) \right) \nabla \alpha_\varphi \end{aligned} \tag{2.68}$$

For what concerns the interfacial drag term it is modelled as follows:

$$\bar{\mathcal{M}}_{h,\varphi} = \lambda(Re_I, \pi_\mu) \frac{|\nabla \alpha_\varphi|}{\delta} \frac{\bar{\mu}^\varphi \bar{\mu}^\phi}{\bar{\mu}^\varphi + \bar{\mu}^\phi} \left( \bar{U}^\varphi - \bar{U}^\phi \right) \tag{2.69}$$

Where $\bar{\mu}$ is the field average viscosity, $|\nabla \alpha_\varphi|$ is used to model $\Sigma$, i.e. the interfacial area density that varies across the interfacial transition region of width $\delta$[3]. $\lambda(Re_I, \pi_\mu)$ is a proportionality factor representing the dimensionless friction coefficient which holds tangential inertia and tangential shear contributions:

$$\lambda(Re_I, \pi_\mu) = mRe_I + n\pi_\mu \tag{2.70}$$

the parameters $m$ and $n$ have to be appropriately chosen ($0.1 \leq m \leq 1.5$ and $n \approx 8$ ). From this, it is evident that the dissipative drag is a function of the

[3]has to be modelled, see [19]. In OpenFOAM is defined as *mesh.V*

relative phase velocity, that is the averaged slip velocity between the phases. Note that the interfacial Reynolds number has been defined as:

$$Re_I \equiv \frac{\rho\delta|\bar{U}^{\varphi} - \bar{U}^{\phi}|}{\alpha_{\varphi}\alpha_{\phi}\bar{\mu}^{\varphi}\bar{\mu}^{\phi}/(\bar{\mu}^{\varphi} + \bar{\mu}^{\phi})} \tag{2.71}$$

Moreover, for the dimensionless group $\pi_{\mu}$ (viscous shear contribution), it is suggested:

$$\pi_{\mu} \equiv \frac{\alpha_{\varphi}\alpha_{\phi}\bar{\mu}^{\varphi}\bar{\mu}^{\phi}/(\alpha_{\phi}\bar{\mu}^{\varphi} + \alpha_{\varphi}\bar{\mu}^{\phi})}{\bar{\mu}^{\varphi}\bar{\mu}^{\phi}/(\bar{\mu}^{\varphi} + \bar{\mu}^{\phi})} \tag{2.72}$$

# Chapter 3

# Implementation in OpenFOAM

In this Chapter, the OpenFOAM implementation of the two-fluid model is described. The discretization of the governing equations in particular of the *MULES::explictiSolve* and the PIMPLE p-v coupling described in *pEqn.H* are listed along with the discretization of momentum, energy and species transport. This last transport equation will be highlighted with the barotropic cavitation term as the substantial difference between the two OpenFOAM solvers considered in this work, namely *twoPhaseEulerFoam* and *reactingTwoPhaseEulerFoam*.

## 1 twoPhaseEulerFoam

Solver for a system of two compressible fluid phases with one phase dispersed, e.g. gas bubbles in a liquid including heat-transfer [47].

*twoPhaseEulerFoam* handles the requirement to describe the properties of the individual phases as well as the interaction terms between the phases by two distinct libraries *phaseModel* and *twoPhaseSystem*. Fig. 3.1 shows the flow chart of the solver, and hence the algorithm followed by twoPhaseEulerFoam to solve the fluid-dynamic problem.
The procedure starts with updating the timestep according to the Courant number limit and then solves the coupled set of volume fraction equations with interface sharpening for selected phase pairs. The interface momentum transfer coefficients are computed and an equation for the phase velocities is constructed and solved for preliminary values. The PIMPLE (PISO + SIMPLE) algorithm is used to solve pressure-velocity coupling and will be briefly described following in the chapter.

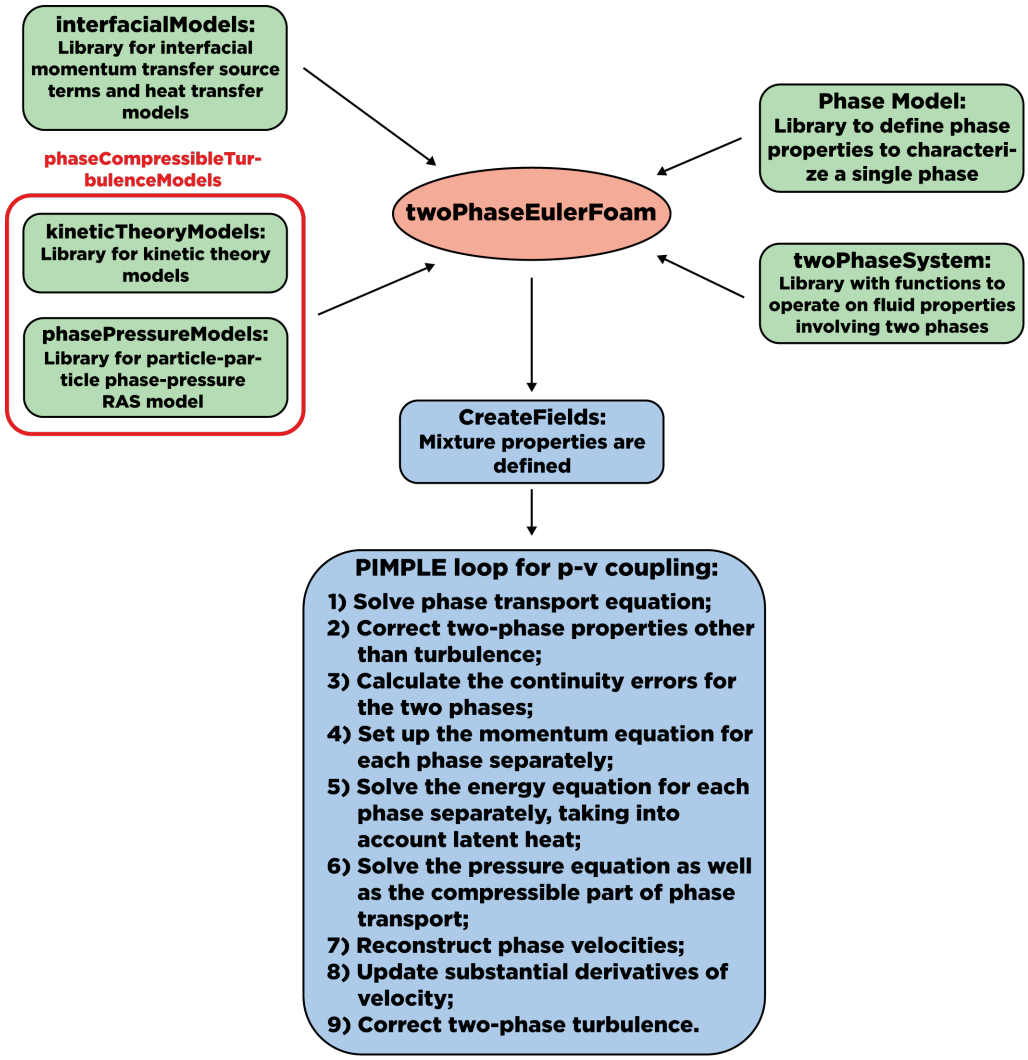


Figure 3.1: Flow chart of *twoPhaseEulerFoam* solver as implemented in OpenFOAM-dev

## 2 reactingTwoPhaseEulerFoam

Solver for a system of two compressible fluid phases with a common pressure, but otherwise separate properties. The type of phase model is run time selectable and can optionally represent multiple species and in-phase reactions. The phase system is also run time selectable and can optionally represent different types of momentum, heat and mass transfer [47].

The solution procedure is similar to the one seen in section 1 but this

solver features two other transport equations to take into account the mass-fraction transport inside each phase. This diversity can be noted in Fig. 3.2 where an additional step is added before the solution of the PIMPLE loop in order to take into account the species transport. As well as the transport equations also a term to take into account the species exchange between phases is needed, namely the *interfaceComposition* library. This, as it will be explained later in this chapter, allowed to identify a framework able to fit a barotropic cavitation model.

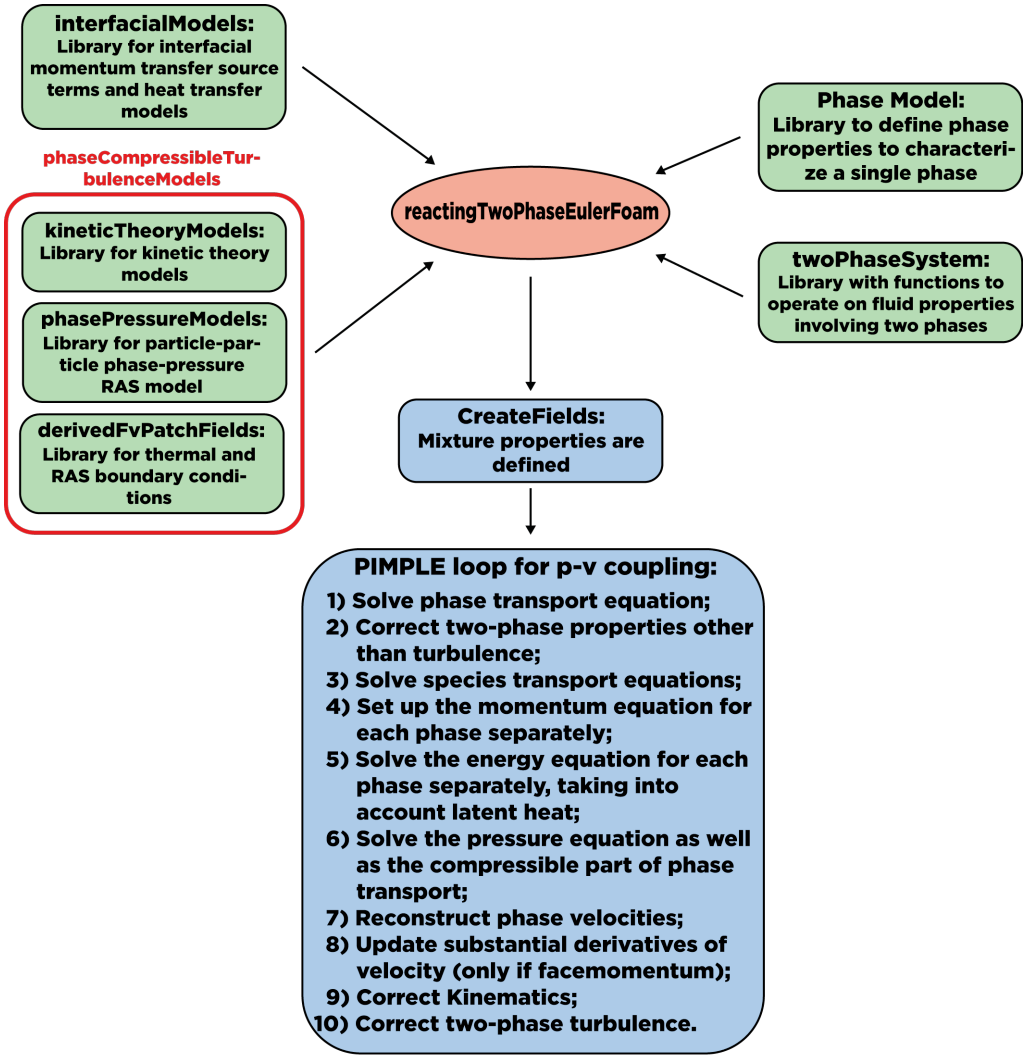


Figure 3.2: Flow chart of *reactingTwoPhaseEulerFoam* solver as implemented in OpenFOAM-dev

## 2.1 The Species Transport Equation

As already said in the introduction of this section, the species transport equations represent one of the main diversities between the two solvers considered in this work. It is, then, important to understand how the solver interprets the mass transfer between phases.

**Listing 3.1: YEqns.H file**

```
{
autoPtr<phaseSystem::massTransferTable>
massTransferPtr(fluid.massTransfer());

phaseSystem::massTransferTable&
massTransfer(massTransferPtr());

PtrList<volScalarField>& Y1 = phase1.Y();
PtrList<volScalarField>& Y2 = phase2.Y();

forAll(Y1, i)
    {
      tmp<fvScalarMatrix> Y1iEqn(phase1.YiEqn(Y1[i]));

      if (Y1iEqn.valid())
      {
          Y1iEqn =
          (
              Y1iEqn
              ==
              *massTransfer[Y1[i].name()]
              + fvOptions(alpha1, rho1, Y1[i])
          );
...
```

The file links to an auto-pointer where the mass transfer rates table are calculated for all species. The pointer is defined in:

`\$FOAM_APP/solvers/multiphase/reactingEulerFoam/phaseSystems/PhaseSystems/InterfaceCompositionPhaseChanghePhaseSystem/InterfaceCompositionPhaseChanghePhaseSystem.C`

This file defines the amount of mass transfer between species/phases. The term referring to the mass transfer rate is given by *dmdt*. In order to calculate this term a mass transfer model is needed. The default ones implemented in OpenFOAM are the *Spherical* and *Frössling* models [48]. Once the mass transfer term is calculated the script substitutes it inside the mass transfer term of the specie equation in the following form:

**Y1iEqn == *massTransfer[Y1[i].name()]+fvOptions(alpha1, rho1, Y1[i])**

where the left hand side is the specie transport equation created upon the phase model selected. In our case is the *multiComponentPhaseModel*, this means that per each specie a Y equation is created as defined in:

`\$FOAM_APP/solversmultiphase/reactingEulerFoam/phaseSystems/phaseModel/MultiComponentPhaseModel/MultiComponentPhaseModel.C`

As it is possible to read in the following code:

**Listing 3.2: Species transport equation as defined in the "BasePhaseModel"**

```
template<class BasePhaseModel>
Foam::tmp<Foam::fvScalarMatrix>
Foam::MultiComponentPhaseModel<BasePhaseModel>::YiEqn
(
...

return
(
      fvm::ddt(alpha, rho, Yi)
      + fvm::div(alphaRhoPhi, Yi, div( +alphaRhoPhi.name()
          + ,Yi))
      - fvm::Sp(this->continuityError(), Yi)
      - fvm::laplacian
      (
        fvc::interpolate(alpha)
        *fvc::interpolate(this->turbulence().nut()*rho/Sc_
            ),
        Yi
      )
      ==
      alpha*this->R(Yi)

      + fvc::ddt(residualAlpha_*rho, Yi)
      - fvm::ddt(residualAlpha_*rho, Yi)
);
```

it is then straightforward the derivation of the specie equation:

$$\frac{\partial}{\partial t}(\alpha\rho Y_i) + \nabla(\alpha\rho\vec{U}Y_i) - \nabla\cdot\left[\alpha\frac{\mu_t}{Sc_t}\nabla Y_i\right] = R \tag{3.1}$$

With:

$\mu_t$: Turbulent viscosity;

$Sc_t$: Turbulent Schmidt number;

$R$: Source term due to reaction;

The mass transfer term has to be added to this formulation of the specie equation in order to have the actual equation that the solver is using, that will be:

$$\frac{\partial}{\partial t}(\alpha\rho Y_i) + \nabla(\alpha\rho\vec{U}Y_i) - \nabla\cdot\left[\alpha\frac{\mu_t}{Sc_t}\nabla Y_i\right] = R + massTransfer[Y_i] \quad (3.2)$$

As we will see in the following section the inter-phase mass transfer in the species equation, since it is a source/sink term, is considered as part of the massTransfer term of the species equation.

## 2.2 The Interface Composition Model

This section will focus on how the interface composition model is used to calculate species composition at the interface in order to take, correctly, into account for species mass transfer at the interface. For starters an extract of the setup file ***constant/phaseProperties*** is presented to understand how the ***interfaceComposition*** call is made:

**Listing 3.3: Template of an interfaceComposition call in phaseProperties**

```
interfaceComposition
(
    (gas in liquid)
    {
        type Saturated;
        species ( H2O );
        Le 1.0;
        saturationPressure
        {
            type Antoine;
            A XXX;
            B XXX;
            C XXX;
        }
    }
);
```

The interface composition model, as defined in ***InterfaceCompositionModel.C*** computes various quantities that are then used by ***interfaceCompositionPhaseChangePhaseSystem.C*** to compute $\frac{dm}{dt}$ and populate the massTransferTable.

First of all it is interesting to focus on the type of interface composition model, hence on the ***Saturated.C*** file. It calculates two variables, namely:

**Yf:** The interface species fraction;

**YfPrime:** The interface species fraction derivative with respect to temperature.

The program firstly identifies whether the species belong to saturated or not, as specified in the listing above ("species ( H2O )"), in this case the specie will be "H2O".

$$Y_f = \begin{cases} \frac{MW_{sat}}{p \cdot MW_{tot}} \cdot p_{sat}(T_f) & \text{if : "}specie = saturated\text{"} \\ \frac{Y_i \cdot \left(1 - \frac{MW_{sat}}{p \cdot MW_{tot}} \cdot p_{sat}(T_f)\right)}{1 - Y_{sat}} & \text{otherwise} \end{cases} \tag{3.3}$$

Where $Y_i$ represents the mass fraction of the "non saturated specie" while $Y_{sat}$ represents the mass fraction of the "saturated specie. As it is possible to infer from the definition of $Y_f$ this is the interface composition model that describes the phase change in the "saturated" specie.

$$y_i \cdot p = p_i = x_i \cdot p_{sat} \tag{3.4}$$

The term $\frac{MW}{p \cdot MW_{tot}}$ in equation 3.3 represents the constant of proportionality between partial pressure and mass, hence $Y_f$ represents the mass fraction of evaporated specie (the interface specie mass fraction). From equation (3.4) it is easily derivable the first expression of equation 3.3, since:

$$Y_f = y_i \frac{MW_i}{MW_{tot}} = \frac{p_{sat_i}(T)}{p} \frac{MW_i}{MW_{tot}} \tag{3.5}$$

The second part of equation 3.3 represents, instead, the interface mass fraction of the other species.
The interface composition, hence, models the sudden jump in the value of gas mass fraction at interface due to gas solubility in liquid. This is not the only component of mass transfer between phases, but two other settings of phaseProperties dictionary had to be defined in order to model the mass transfer coefficient through either Frossling or Spherical [48]. For each phase to be transported two mass transfer coefficient are defined, in order to distinguish whether the liquid or vapour are a continuous or dispersed phase. In other words, if we focus on the mass transfer of the gaseous phase, two different mas transfer coefficient will be defined: one to describe the mass transfer of gas as a diluted phase inside the liquid and another one to model

a continuous gas with liquid as diluted phase. The same is done for the other phase, in this case the liquid one.
Due to the presence of discontinuities in concentration at phase equilibrium, it is, in general, not possible to model multicomponent mass transfer using a single overall mass transfer coefficient. Instead, it is necessary to consider a generalization of the heat transfer two resistances model. For instance, consider one specie ($i$) dissolved in two immiscible phases **A** and **B**. The basic assumption of this model is that there is no resistance to mass transfer at the interface, and hence the equilibrium conditions prevail at the interface (namely $Y_f$). The specie mass transfer is modelled through the coefficients $K_A$ and $K_B$ defined on either side of the phase interface, these are defined so that driving forces are defined in terms of mass fraction differences. Thus the mass flux of specie $i$ to phase **A** from the interface is:

$$\dot{m}_{i_A} = \rho_i K_A (Y_{f_A} - Y_{i_A}) \tag{3.6}$$

And the mass flux of $i$ to phase **B** from the interface is:

$$\dot{m}_{i_B} = \rho_i K_B (Y_{f_B} - Y_{i_B}) \tag{3.7}$$

A graphical representation of this model is presented in figure 3.3, in which is possible to highlight the necessity of two mass transfer coefficients and also highlights the discontinuity at interface due to the phase change.

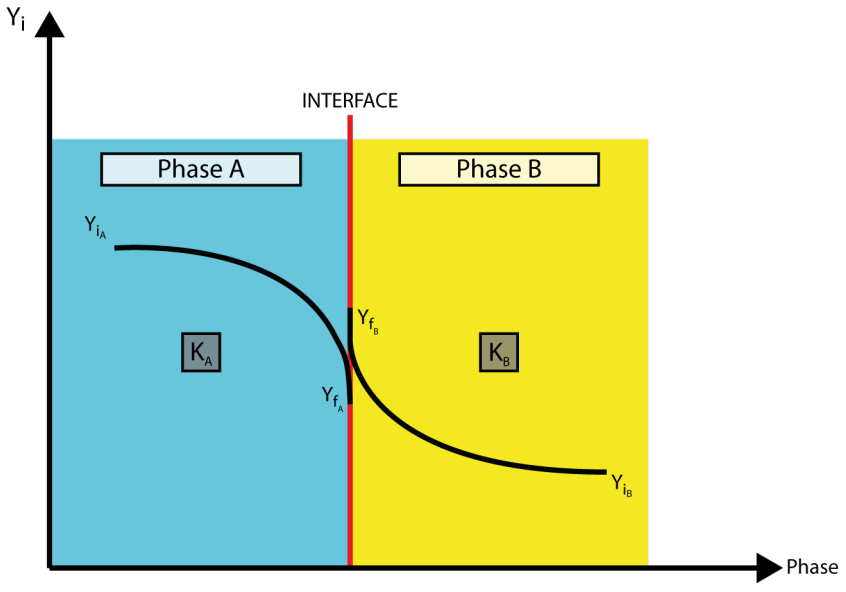


Figure 3.3: Interface species transfer with phase change

This is implemented in OpenFOAM through ***interfaceCompositionPhaseChangePhaseSystem.C*** code.

### 2.2.1 Saturation pressure modelling

Several saturation models are implemented, and each of them outputs the same variables ($P_{sat}$, $P'_{sat}$, $ln(P_{sat})$ and $T_{sat}$). They differ in the way these properties are calculated, in detail we have:

- *Antoine* [49], it uses Antoine's equation to derive the saturation pressure and needs the calibration of three coefficients (A, B, C):

$$log(P_{sat}) = A + \frac{B}{C - T}; \tag{3.8}$$

- *Antoine Extended*, it uses a modified Antoine's equation to derive the saturation pressure and needs the calibration of six coefficients (A, B, C, D, E, F):

$$log(P_{sat}) = A + \frac{B}{C - T} + Dlog(T) + ET^F; \tag{3.9}$$

- *ArdenBuck* [50], It is based on the vapour pressure of moist air;
- *constant*, It is based on constant saturation conditions;
- *Function1*, It express saturation vapour temperature in terms of pressure of saturation, the inverse function is not implemented;
- *Polynomial*, It describes saturation temperature as a polynomial function of saturation pressure

$$T_{sat} = \sum C^i P^i_{sat} \tag{3.10}$$

As already stated, the most suitable model consist in the *Antoine*, this model in fact describes easily the saturation pressure as a function of temperature and also is a reliable and highly validated model.

The focus of this analysis is then moved to the file *InterfaceCompositionModel.C*, in which are defined the following variables:

$$dY = Y_f - Y_i \tag{3.11}$$

That represents the mass difference between the fluid interface and the field.

$$D = \frac{\alpha_H}{\rho Le} \tag{3.12}$$

That defines the mass diffusivity as a function of thermal diffusivity ($\alpha_H$) and Lewis number ($Le$).

$$L = Ha_1 - Ha_2 \tag{3.13}$$

Also the latent heat of evaporation/condensation is defined as the difference of the absolute enthalpy of the two phases. And finally the heat flux of evaporation/condensation and its derivative are calculated:

$$\dot{m}L = \rho KDL \cdot dY \tag{3.14}$$

$$\dot{m}'L' = \rho KDL \cdot Y_f' \tag{3.15}$$

Finally, moving to the file *interfaceCompositionPhaseChangePhaseSystem.C*, all the contributions from each interface composition model are summed inside the *massTransferTable*. The most interesting variables calculated are $eqns_1$ and $eqns_2$, representing respectively the generation/consumption of specie i for both phase 1 and 2, as it is possible to see in figure 3.4.

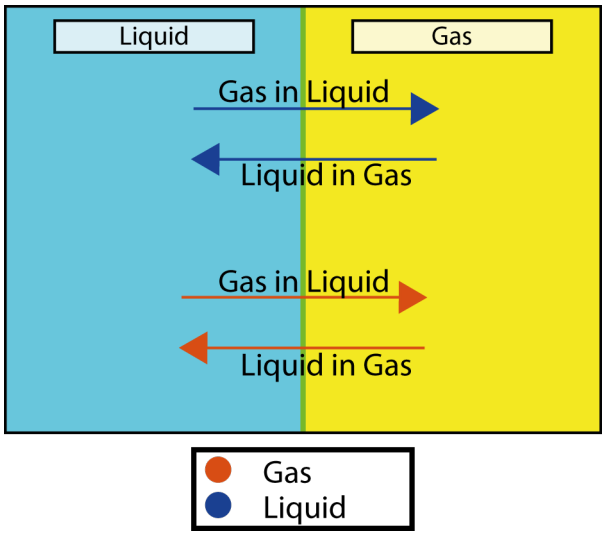


Figure 3.4: Phase change of dispersed phase

These two arrays are populated following the criterion of "dispersed" and "continuous" phase, or rather, since the script iterates over each interface composition model and since each of those is defined as either "*gas in liquid*" or "*liquid in gas*" through ***orderedPhasePair.C*** it is possible to identify which specie is continuous and which is dispersed. In other words, each model is defined for an ordered phase pair, eg. gas in liquid, this means that, as in our case, the gas is the dispersed phase while the liquid is the continuous one. This definition allows a correct allocation of the mass transferred through the

phases, since the mass transfer table for each specie is defined as:

$$eqns[phase1] = \rho_1 KD \cdot Y_f - Sp(\rho_1 KD\psi) \tag{3.16}$$

$$eqns[phase2] = \rho_2 KD \cdot dY \tag{3.17}$$

Where "phase1" and "phase2" refer respectively to the dispersed and continuous phase. For instance, if there is gas generation (according to the "Saturated" model) a gas mass increase will be accounted through eqns[phase1] while a liquid mass decrease will be determined through eqns[phase2], this will not happen to a inert specie that, as it is possible to derive from equations 3.16 and 3.17 will be transported in the gas phase but will have a null $dY$ and , hence no liquid mass generation/destruction. In order to better understand a graphical representation of this process is presented in figure 3.5.

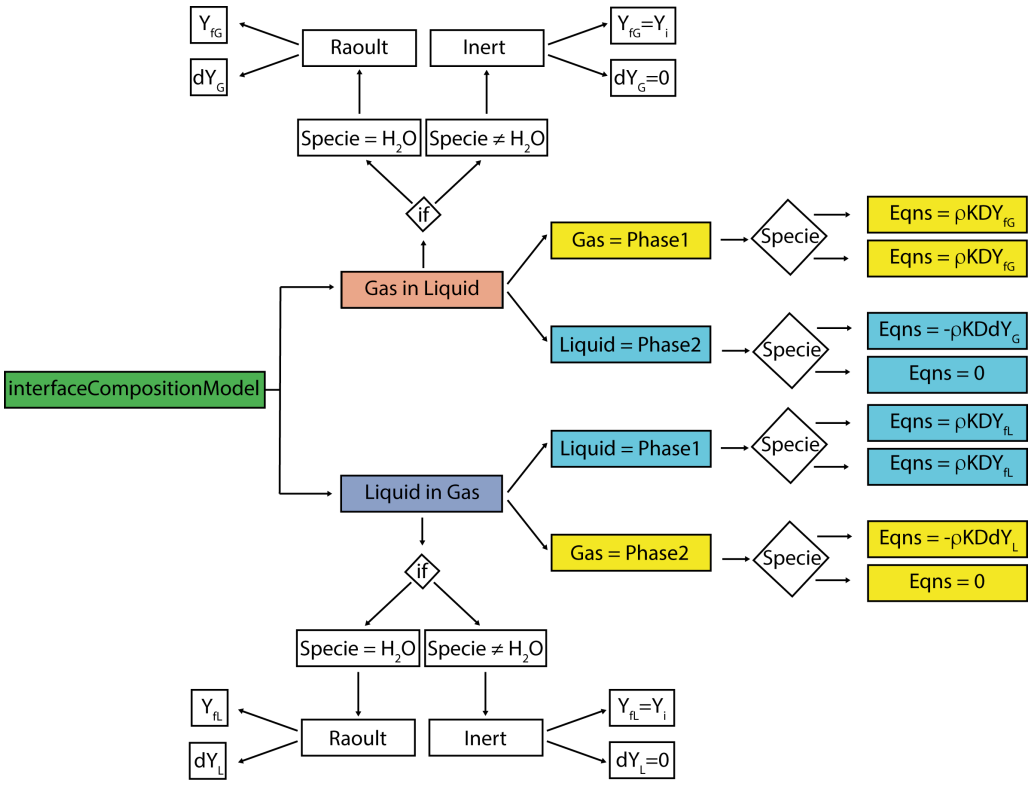


Figure 3.5: OpenFOAM procedure to determine phase change and mass transfer.

From this scheme it is also possible to infer that a model based on Antoine equation is needed in order to describe cavitation and condensation. This formulation of *interfaceCompositionPhaseChangePhaseSystem.C* generates also

the variable $dm/dt$, that is used to calculate the source term of the phase equation. This quantity is defined through equation (3.16) and expressed as its two components the former explicit and the latter implicit:

$$\left(\frac{dm}{dt}\right)_{explicit} = (-1)^k KD \cdot Y_f \tag{3.18}$$

$$\left(\frac{dm}{dt}\right)_{implicit} = -(-1)^k KD\psi \tag{3.19}$$

Where "k" is either 0 or 1 depending whether the mass is generated or destructed. The study of this interface composition model allowed the author to identify a structure of the code able to welcome the cavitation model. It is, hence easy to substitute theoretically the *"Saturated"* model with a simple barotropic one using the Antoine's equation to describe the saturation pressure at different temperatures. This will allow the analytical description of the source terms due to cavitation in the species equation. Later on in this chapter will be highlighted that this source term will determine a change in the phase fraction representing the sink/source term for the phase fraction equation due to cavitation.

### 2.2.2 Thermo correction due to latent heat

interfaceCompositionPhaseChangePhaseSystem.C has also a section dedicated to the temperature correction this loop solves for the interface temperatures, Tf, and updates the interface composition models. The correction is based on the fact that the rate of heat transfer to the interface must equal the latent heat consumed at the interface, i.e.:

$$h_1(T_1 - T_f) + h_2(T_2 - T_f) = \dot{m}L = K\rho_i \cdot L_i(Y_{f_i} - Y_i) \tag{3.20}$$

Where $h_1$ and $h_2$ are respectively the convective heat transfer coefficient of phase 1 and phase 2. From equation (3.20) it is easily derivable the corrected interface temperature, that will be used to derive the new cavitated/condensed mass fraction.

# 3 PIMPLE Loop

After the introduction of the two solvers and the focus on the species mass fraction transport equation it is time to detail the PIMPLE loop. Firstly a brief introduction on what it represents for the solution algorithm is presented and then, in the following sections a description of the discretization and solution process will be carried out for the most important equations, namely the transport equations of the two-phase system described in chapter 2.

Most fluid dynamics solver applications in OpenFOAM use either the pressure-implicit split-operator (PISO), the semi-implicit method for pressure-linked equations (SIMPLE) algorithms, or a combined PIMPLE algorithm. These algorithms are iterative procedures for coupling equations for momentum and mass conservation, PISO and PIMPLE being used for transient problems and SIMPLE for steady-state. Within in time, or solution, step, both algorithms solve a pressure equation, to enforce mass conservation, with an explicit correction to velocity to satisfy momentum conservation. They optionally begin each step by solving the momentum equation, the so-called momentum predictor. While all the algorithms solve the same governing equations (albeit in different forms), the algorithms principally differ in how they loop over the equations. The looping is controlled by input parameters that are listed below. They are set in a dictionary named after the algorithm [47].

- *nCorrectors*:used by PISO, and PIMPLE, sets the number of times the algorithm solves the pressure equation and momentum corrector in each step; typically set to 2 or 3.
- *nNonOrthogonalCorrectors*:used by all algorithms, specifies repeated solutions of the pressure equation, used to update the explicit non-orthogonal correction of the Laplacian term $\nabla \cdot \frac{1}{\mathcal{A}} \nabla p$; typically set to 0 (particularly for steady-state) or 1.
- *nOuterCorrectors*:used by PIMPLE, it enables looping over the entire system of equations within on time step, representing the total number of times the system is solved; must be $\geq 1$ and is typically set to 1, replicating the PISO algorithm.
- *momentumPredictor*:switch that controls solving of the momentum predictor; typically set to off for some flows, including low Reynolds number and multiphase.

# 4 The Alpha Equation

The phase fraction transport equation is the first equation called in the PIMPLE loop. It is technically solved by the MULES algorithm, but actually the fields needed by it are created in the `twoPhaseSystem.C` file. This is why this section will focus on how the fields are managed in OpenFOAM and only in the following section the actual discretization of the alpha equation will be analyzed. For simplicity the analysis will focus on *reactingTwoPhaseEulerFom*, it will hence be described the reacting solver with the source mass term derived from the species mass transport equation as part of both explicit and implicit source term of the void fraction transport equation.

## 4.1 The Dilatation Rate Term

Actually the first step implemented in `twoPhaseSystem.C` is to define the dilatation rate source term, it consists basically in the source term of alpha equation due to compressibility, but in the reacting case it is also enriched with the $dm/dt$ term of `"YEqns.H"` as said before.

### 4.1.1 The Effect of Compressibility

It is now possible to identify the definition of the phase dilatation rate:

**Listing 3.4: The phase dilatation rate in `twoPhaseSystem.C`**

```
if (phase1_.divU().valid() && phase2_.divU().valid())
{
tdgdt =
(
alpha2()
*phase1_.divU()()()
- alpha1()
*phase2_.divU()()()
);
}
```

In order to find a definition for ***phase1.divU()()()*** the file `pEqn.H` is needed to be analyzed.
Here the two phase dilatation rate are defined as function of the compressible part of the pressure equation [see Appendix A for the computation of *divU*]:

**Listing 3.5: Definition of *divU* in `pEqn.H`**

```
// Set the phase dilatation rates
if (pEqnComp1.valid())
{
phase1.divU(-pEqnComp1 & p_rgh);
}
if (pEqnComp2.valid())
{
phase2.divU(-pEqnComp2 & p_rgh);
}
```

So, finally it is possible to derive the elementary component of ***dgdt***:

**Listing 3.6: Compressible part of the pressure equation for phase 1 in `pEqn.H`**

```
{
pEqnComp1 =
(
phase1.continuityError()
- fvc::Sp(fvc::ddt(alpha1) + fvc::div(alphaPhi1), rho1)
)/rho1
+ (alpha1*psi1/rho1)*correction(fvm::ddt(p_rgh));
}
```

Now a complete analytical formulation of the phase dilatation rate will be derived starting from the OpenFOAM formulation.

$$\frac{dg}{dt} = \alpha_1 divU_{\alpha_2} - \alpha_2 divU_{\alpha_1} \tag{3.21}$$

$$divU_{\alpha_\varphi} = -pEqComp_{\alpha_\varphi} \tag{3.22}$$

$$pEqnComp_{\alpha_\varphi} = -\left(\frac{\partial \alpha_\varphi}{\partial t} + \nabla \cdot (\alpha_\varphi \vec{U}_\varphi)\right) + \frac{\alpha_\varphi \psi_\varphi}{\rho_\varphi}\frac{D(p_{rgh})}{Dt}(1-\Psi) \tag{3.23}$$

Here $\Psi$, is an underelaxation factor introduced to obtain boundedness in the solution. It will be kept in the formulation of the $\frac{dg}{dt}$ and erased only when fully made explicit.
Now equations (3.23) and (3.22) can be substituted into (3.21) to lead:

$$\begin{aligned}\frac{dg}{dt} = \alpha_1 &\left[\frac{\partial \alpha_2}{\partial t} + \nabla \cdot (\alpha_2 \vec{U}_2) - \frac{\alpha_2 \psi_2}{\rho_2}\frac{D(p_{rgh})}{Dt}(1-\Psi)\right] + \\ - \alpha_2 &\left[\frac{\partial \alpha_1}{\partial t} + \nabla \cdot (\alpha_1 \vec{U}_1) - \frac{\alpha_1 \psi_1}{\rho_1}\frac{D(p_{rgh})}{Dt}(1-\Psi)\right]\end{aligned} \tag{3.24}$$

(3.24) can be rearranged as:

$$\begin{aligned}\alpha_1\left[\frac{\partial\alpha_2}{\partial t}+\nabla\cdot(\alpha_2\vec{U}_2)-\frac{\alpha_2\psi_2}{\rho_2}\frac{D(p_{rgh})}{Dt}(1-\Psi)\right]+\\ -(1-\alpha_1)\left[\frac{\partial\alpha_1}{\partial t}+\nabla\cdot(\alpha_1\vec{U}_1)-\frac{\alpha_1\psi_1}{\rho_1}\frac{D(p_{rgh})}{Dt}(1-\Psi)\right]\end{aligned} \tag{3.25}$$

This formulation can be expanded as follows:

$$\begin{aligned}\alpha_1\frac{\partial\alpha_1}{\partial t}+\alpha_1\nabla\cdot(\alpha_2\vec{U}_2)-\frac{\alpha_1\alpha_2\psi_2}{\rho_2}\frac{Dp_{rgh}}{Dt}(1-\Psi)-\frac{\partial\alpha_1}{\partial t}+\\ -\nabla\cdot(\alpha_1\vec{U}_1)+\frac{\alpha_1\alpha_2\psi_1}{\rho_1}\frac{Dp_{rgh}}{Dt}(1-\Psi)+\alpha_1\frac{\partial\alpha_1}{\partial t}+\alpha_1\nabla\cdot(\alpha_1\vec{U}_1)\end{aligned} \tag{3.26}$$

Remembering that:

$$\alpha_1\frac{\partial\alpha_1}{\partial t}+\alpha_1\frac{\partial\alpha_2}{\partial t}=0 \tag{3.27}$$

$$\alpha_1\nabla\cdot(\alpha_1\vec{U}_1)+\alpha_1\nabla\cdot(\alpha_2\vec{U}_2)=\alpha_1\nabla\cdot(\vec{U}) \tag{3.28}$$

The rearranged equation becomes:

$$\alpha_1\alpha_2\frac{Dp_{rgh}}{Dt}\left(\frac{\psi_1}{\rho_1}(1-\Psi)-\frac{\psi_2}{\rho_2}(1-\Psi)\right)-\left(\frac{\partial\alpha_1}{\partial t}+\nabla\cdot(\alpha_1\vec{U}_1)-\alpha_1\nabla\cdot(\vec{U})\right) \tag{3.29}$$

Then it is simplified through equations (2.22) and (2.19), to lead:

$$\begin{aligned}\underbrace{\alpha_1\alpha_2\left(\frac{\psi_2}{\rho_2}\frac{Dp_{rgh}}{Dt}(1-\Psi)-\frac{\psi_1}{\rho_1}\frac{Dp_{rgh}}{Dt}(1-\Psi)\right)}_{n+1}+\\ +\underbrace{\alpha_1\alpha_2\left(\frac{1}{\rho_2}\frac{D\rho_2}{Dt}-\frac{1}{\rho_1}\frac{D\rho_1}{Dt}\right)}_{n}\end{aligned} \tag{3.30}$$

According to Weller [14] in the solver the pressure equation is constructed using the corrected flux prediction (n+1), while the calculation of the total derivatives of the densities in the compressibility terms are performed using the fluxes evaluated at the previous iteration (i.e. before the flux prediction, n).
It is now possible to focus on the ***bounding*** term, which can be correctly written as:

$$\alpha_1\alpha_2\left[\frac{\psi_2}{\rho_2}\left[\frac{Dp^n_{rgh}}{Dt}-\frac{Dp^{n-1}_{rgh}}{Dt}\right]-\frac{\psi_1}{\rho_1}\left[\frac{Dp^n_{rgh}}{Dt}-\frac{Dp^{n-1}_{rgh}}{Dt}\right]\right] \tag{3.31}$$

As it is possible to see from (3.31) the bounding term approaches zero as the solver approaches convergence. It is also important to highlight that this term has no physical meaning and, hence, it will not be considered in the definition of the $\frac{dg}{dt}$. It is finally possible to derive the final formulation of $\frac{dg}{dt}$ as:

$$\frac{dg}{dt} = \alpha_1\alpha_2\left(\frac{1}{\rho_2}\frac{D\rho_2}{Dt} - \frac{1}{\rho_1}\frac{D\rho_1}{Dt}\right) \tag{3.32}$$

From equation (3.32) is possible to see that the dimensions of the phase dilatation rate are consistent with the formulation of the mules explicit solver. And also the $\frac{dg}{dt}$ represents the source term due to compressibility of equation (2.22).

### 4.1.2 The Mass Transfer Term

Now the solver adds the terms due to the mass transfer, simply by considering the source term in terms of volume generated/destructed through the mass generated/destructed per unit time divided by its density, namely:

**Listing 3.7: The compressibility contribution due to mass transfer to pressure and alpha equation**

```
if (fluid.transfersMass())
    {
        if (pEqnComp1.valid())
        {
            pEqnComp1.ref() -= fluid.dmdt()/rho1;
        }
        else
        {
            pEqnComp1 = fvm::Su(-fluid.dmdt()/rho1, p_rgh);
        }

        if (pEqnComp2.valid())
        {
            pEqnComp2.ref() += fluid.dmdt()/rho2;
        }
        else
        {
            pEqnComp2 = fvm::Su(fluid.dmdt()/rho2, p_rgh);
        }
    }
```

This is the demonstration that the phase fraction and mass fraction transport equations are linked, and that the change of phase of a specie would be tracked in the void fraction equation as a source/sink term as we will see briefly in the following paragraph.

## 4.2 The Alpha Equation Source Terms

Following, in `twoPhaseSystem.C` the implicit and explicit source terms *Sp* and *Su* are initialized and calculated:

$$Sp^n = \begin{cases} 0 & \text{if} \quad n = 0 \\ Sp^{n-1} - \frac{\partial g}{\partial t}\frac{1}{\alpha_2} & \text{if} \quad \frac{\partial g}{\partial t} > 0 \\ Sp^{n-1} + \frac{\partial g}{\partial t}\frac{1}{\alpha_1} & \text{if} \quad \frac{\partial g}{\partial t} < 0 \end{cases}$$

$$Su^n = \begin{cases} \alpha_1 div(\phi) & \text{if} \quad n = 0 \\ Su^{n-1} + \frac{\partial g}{\partial t}\frac{1}{\alpha_2} & \text{if} \quad \frac{\partial g}{\partial t} > 0 \end{cases}$$

It is important to remember the definition of the phase dilatation rate $\left(\frac{dg}{dt}\right)$ as well as the definition of the fluxes that will be showed in the following paragraph.

## 4.3 The Fluxes

In order to solve the phase transport equation, a combined flux is derived to assure boundedness. Instead of writing the flux directly from the phase velocity, provided directly by the solution of the phase intensive momentum equation, the flux for phase $\varphi$ is calculated as linear combination of the absolute velocity and the relative velocity as seen in the previous chapter (equations 2.16 and 2.17). This leads to the definition of two fluxes, one relative to the absolute velocity and the other derived from the relative one. So that the "$alphaPhi_\varphi$" field, namely the phase intensive flux needed by the MULES algorithm to solve explicitly the void fraction transport equation is calculated as follows:

$$\alpha_\varphi \phi_\varphi = \alpha_\varphi \phi + \alpha_\varphi (1 - \alpha_\varphi)\phi_r \tag{3.33}$$

However, the solution is not that simple since the relative flux is not only calculated as the difference between the fluxes derived from the two different phases, but also includes a correction term. The formulation will hence be:

$$\phi_r = \frac{\bar{D}}{A}\nabla\alpha \cdot S_f + (\phi_1 - \phi_2) \tag{3.34}$$

Where $\frac{\bar{D}}{A} = \left[\left(\frac{D}{A}\right)_1 + \left(\frac{D}{A}\right)_2\right]$.
Now the first addend of equation (3.34) represents the correction term. The reader is suggested to reference the work of Rousche [15] for the analytical derivation of the correction term. Hereby a brief treatment of the problem is

presented, remembering that the main difference between this work and the one carried by Rousche is the compressibility of the phases.

From [15] it is possible to derive the momentum correction, as Rousche mentioned in his work this term derives from a semi-discretised form of the momentum equation:

$$\left(\mathcal{A}_\varphi\right)_D \vec{U}_\varphi = \left(\mathcal{A}_\varphi\right)_H - \nabla\bar{p} - \frac{A_\alpha \nabla\alpha_\varphi}{\alpha_\varphi} \tag{3.35}$$

Where $\mathcal{A}_\varphi$ denotes the system of linear algebraic equations arising from the discretisation of the phase momentum equations without terms which are proportional to the gradients $\nabla\bar{p}$ and $\nabla\alpha_\varphi$. Analogously at Rhie and Chow [51] procedure, The gradient terms are not discretised at this stage. Rearranging equation (3.35) we obtain the momentum correction equation:

$$\vec{U}_\varphi = \frac{\left(\mathcal{A}_\varphi\right)_H}{\left(\mathcal{A}_\varphi\right)_D} - \frac{\nabla\bar{p}}{\left(\mathcal{A}_\varphi\right)_D} - \frac{A_\alpha \nabla\alpha_\varphi}{\alpha_\varphi\left(\mathcal{A}_\varphi\right)_D} \tag{3.36}$$

Once the momentum correction equation (3.36) is derived, it is possible to substitute it in equation (2.22):

$$\frac{\partial\alpha_1}{\partial t} + \nabla\cdot\left(\frac{\left(\mathcal{A}_1\right)_H}{\left(\mathcal{A}_1\right)_D}\right) - \nabla\cdot\left(\frac{1}{\left(\mathcal{A}_1\right)_D}\nabla\bar{p}\right) - \nabla\cdot\left(\frac{A_\alpha}{\left(\mathcal{A}_1\right)_D}\nabla\alpha_1\right) = 0 \tag{3.37}$$

The problems arise from the last term, i.e. fourth term on the l.h.s., since it is not discretised implicitly. A simple solution is to treat this term implicitly as a diffusion term. However, this remedy suffers from some deficiencies in the sense that the flux featuring in the second term is non-conservative and, consequently, boundedness of the solution can not be guaranteed. This situation can be overcome by starting from equation 2.14 instead of 2.11.
Furthermore, the discretization of the second and third term on the l.h.s. of equation (3.37) can be avoided by correcting the phase velocities for the contributions of the turbulent drag term which are now treated implicitly in the diffusion term. The corrected phase velocity is:

$$\vec{U}^*_\varphi = \vec{U}_\varphi + \frac{A_\alpha}{\left(\mathcal{A}_\varphi\right)_D}\nabla\alpha_\phi \tag{3.38}$$

Substitution of this corrected velocity into equation 2.14, leads to:

$$\begin{aligned}\frac{\partial\alpha_1}{\partial t} &+ \nabla\cdot\left(\vec{U}_1\right) + \nabla\cdot\left(\vec{U}^*_r\alpha_1(1-\alpha_1)\right)\\ &- \nabla\cdot\left[A_\alpha\left(\frac{\alpha_2}{\left(\mathcal{A}_1\right)_D} + \frac{\alpha_1}{\left(\mathcal{A}_2\right)_D}\right)\right] = 0\end{aligned} \tag{3.39}$$

With:

$$\vec{U}_r^* = \vec{U}_1^* - \vec{U}_2^* \tag{3.40}$$

The discretisation of equation (3.39) is:

$$\frac{\alpha_{1_P}^n - \alpha_{1_P}^{n-1}}{\Delta t} V_P + \sum_f \phi \alpha_{1f(\phi,S)}^n + \sum_f \phi_{r_1}^* \alpha_{1_{f(\phi_{r_1}^*,S)}}^n - \sum_f \nu_{\alpha_f} S \cdot \nabla_f \alpha_1^n = 0 \tag{3.41}$$

Where:

- The relative, corrected flux for phase 1 is

$$\phi_{r_1}^* = \alpha_{2_{f(-\phi_{r_1}^*,S)}} \phi_r^* \tag{3.42}$$

- The diffusion coefficient evaluated at the faces is

$$\nu_{\alpha_f} = A_{\alpha_f} \left[ \alpha_{2_f} \left( \frac{1}{\left(\mathcal{A}_1\right)_D} \right)_f + \alpha_{1_f} \left( \frac{1}{\left(\mathcal{A}_2\right)_D} \right)_f \right] \tag{3.43}$$

- The corrected relative flux is calculated by adding the contribution of the turbulent drag term from the volumetric phase fluxes

$$\phi_r^* = \phi_r + S \frac{\nu_{\alpha_f}}{\alpha_{1_f} \alpha_{2_f}} \nabla_f \alpha_1 \tag{3.44}$$

It is also important to notice that in the second term, $\alpha_1$ is bounded at zero as well as one, since the volumetric mixture flux $\phi$ satisfies the mixture continuity equation exactly. In the third term, bounding of $\alpha_1$ is achieved by using $\phi_r^*$ in the convection scheme to interpolate $\alpha_1$ to the face and $-\phi_r^*$ in the face interpolation of $\alpha_2$ . This treatment is quite diffusive if upwind discretisation is used. However, using an higher order differencing scheme instead, reduces the numerical diffusion, but might compromise the boundedness of the solution.

# 5 MULES

The Multidimensional Universal Limiter with Explicit Solution, namely the MULES is an iterative algorithm to solve hyperbolic equations. This method is based on solving the phase fraction equation using an explicit time integration scheme and a combination of low and high order schemes for the fluxes reconstruction that ensure boundedness, mass conservation and sharp interfaces capturing.

## 5.1 ExplicitSolve

The *explicitSolve* function defined in `MULESTemplates.C` is called in the file `twoPhaseSystem.C` to solve the phase fraction equation, this call is made through the command "*fluid.solve()*" as seen in both figure **??** and 3.2. With the hypothesis of no alpha subcycles the function *MULES::explicitSolve* is called as follows:

**Listing 3.8:** ***MULES::explicitSolve*** **call in** `twoPhaseSystem.C`

```
MULES::explicitSolve
(
    geometricOneField(),
    alpha1,
    phi_,
    alphaPhi1,
    Sp,
    Su,
    phase1_.alphaMax(),
    0
);
```

The first argument passed to the constructor is the density, it is passed as *geometricOneField()* which is a unit value field. The second argument is the variable to be solved i.e. the phase fraction. The third argument is the mixture velocity flux, the limited normal convective flux is the next argument which is solved explicitly by the *MULES:limit* function, that will be described in the following section. The next two terms are the explicit and implicit source terms in the continuity equation which arise when mass transfer across the phases or reaction source terms exist. The MULES algorithm solves for the phase fraction with the explicit consideration of the convective flux of the phase fraction. The considered transport mechanism is convection only. If the number of correctors are larger than one, then the phase fraction at the old time is stored and fluxes for all phases are set to zero. The discretized phase transport equation here solved can be written as:

$$\frac{\alpha_{i_P}^{n+1} - \alpha_{i_P}^{n}}{\Delta t} V_p + \sum_f F_{c_f}^n = \alpha_{i_P}^{n+1} S_p + S_u \tag{3.45}$$

Where the subscript "$p$" represent the centroid of a control volume, $f$ represents the faces of the control volume, $F_c$ is the convective flux that is calculated through the function limit, implemented as a part of MULES and the terms $S_p$ and $S_u$ take into account both compressibility and mass transfer source/sink terms as was derived in the previous sections.

## 5.2 Limit

The boundedness of the temporal solution can be achieved with phase fraction and velocity face values limiting schemes such as TVD/NVD convective schemes or by limiting the total convective face fluxes. The second option is implemented in the MULES algorithm, following this criterion the convective face fluxes will split into a low order and high order contribution, to later build a limiter that will allow to estimate the maximum contribution of the high convective fluxes. Convective face fluxes will be computed through:

1. Compute low order convective fluxes $F_C^L$ using upwind scheme for the phase fraction face value reconstruction, operation that ensures boundedness;

2. Compute high order convective fluxes $F_C^H$ using an high order scheme for face phase fraction reconstruction;

3. compute the antidiffusive flux $A = F_C^H - F_C^L$;

4. The convective flux is computed assembling the two previous with the aid of a limiter $\lambda$;

$$F_C = \lambda F_C^H + (1-\lambda)F_C^L = F_C^L + \lambda A \tag{3.46}$$

5. Choose a limiter that ensures boundedness of the solution while retaining the highest order possible of accuracy

From that follows that the most important parameter to be determined in the MULES algorithm is the limiter value at each face. Where a value of the limiter $\lambda = 1$ means that a full high order scheme will be used while $\lambda = 0$ an upwind diffusive scheme will be used to reconstruct face fluxes.

In order to understand the influence of the limiter $\lambda$ in the phase fraction solution it is helpful to rewrite the discretized phase fraction equation (3.45), without considering source terms, adding the antidiffusive convective fluxes:

$$\frac{\alpha_{i_P}^{n+1} - \alpha_{i_P}^{n}}{\Delta t} V_P + \sum_f F_{Cf}^{n,L} + \sum_f \lambda_f A_f^n = 0 \tag{3.47}$$

To calculate the values of the limiters a conservative criterion is used to ensure boundedness of the solution: Supposing that a cell P, bounded by different neighbours NB, has a phase fraction at a given timestep $\alpha_{i_P}^n$, while the neighbours present a value $\alpha_{NB}^n$ for the same property at the exact same timestep. It is therefore possible to demonstrate that the maximum value of

the phase fraction at cell P at the next timestep cannot be bigger then the maximum phase fraction of the neighbouring cells at the previous timestep. The same can be demonstrated for the minimum phase fraction, both sentences are demonstrated through the conservation properties of the transport equation. The convective fluxes' effect on the transport equation reflect in a modification of the phase fraction from a timestep to another and, therefore the contribution of the antidiffusive fluxes must be limited to ensure boundedness. It is, hence, well known that high order convective fluxes offer a less diffusive solution at the expense of boundedness. To better understand the effect of limiters, it is wise to rewrite (3.47) without the contribution of antidiffusive flux:

$$\frac{\alpha_{i_P}^{n+1,L} - \alpha_{i_P}^{n}}{\Delta t} V_P + \sum_f F_{Cf}^{n,L} = 0 \tag{3.48}$$

From (3.48) it is possible to explicit $\alpha_{i_P}^{n+1,L}$, namely the low order approximation of the phase fraction:

$$\alpha_{i_P}^{n+1,L} = \alpha_{i_P}^{n} - \frac{\Delta t}{V_P}\left[\sum_f F_{Cf}^{n,L}\right] \tag{3.49}$$

Subtracting then equations (3.47) and (3.48) leads to:

$$\frac{\alpha_{i_P}^{n+1} - \alpha_{i_P}^{n+1,L}}{\Delta t} V_P + \sum_f \lambda_f A_f^n = 0 \tag{3.50}$$

from which it is possible to highlight the role of antidiffusive fluxes. They represent the difference between low order and highest order possible solution of the phase fraction equation.

Moreover, as stated before in the paragraph, the maximum and minimum value of the phase fraction in the cell P are bounded by their value in the cell and also in neighbouring ones, it is therefore useful to define:

$$\alpha_{P_{max}} = max(\alpha_{i_P}^n, \alpha_{i_{NB}}^n) \tag{3.51}$$

$$\alpha_{P_{min}} = min(\alpha_{i_P}^n, \alpha_{i_{NB}}^n) \tag{3.52}$$

It is also important to take into account numerical issues that may arise during the solution procedure, which may lead to a maximum or minimum phase fraction outside the physical interval $[0, 1]$. Therefore a connection to ensure boundedness has to be performed, following equations (3.51) and (3.52):

$$\alpha_{P_{max}} = max(\alpha_{i_P}^n, 1) \tag{3.53}$$

$$\alpha_{P_{max}} = max(\alpha_{i_P}^n, 0) \tag{3.54}$$

Recalling equation (3.50) it is physically correct to say that positive antidiffusive fluxes lowers the phase fraction while negative ones result in a phase fraction increase. This can be schematically summarized as:

- If $A_f^+ > 0$ then $\alpha_{i_P} \downarrow$
- If $A_f^- < 0$ then $\alpha_{i_P} \uparrow$

Moreover, since phase fraction is limited both superiorly and inferiorly it is possible to derive a condition to which the antidiffusive flux must obey in both cases that arise from equation (3.50):

$$\frac{\alpha_{P_{max}} - \alpha_{i_P}^{n+1,L}}{\Delta t} V_P \geq -\sum_f \lambda_f^- A_f^- \tag{3.55}$$

$$\frac{\alpha_{P_{min}} - \alpha_{i_P}^{n+1,L}}{\Delta t} V_P \leq -\sum_f \lambda_f^+ A_f^+ \tag{3.56}$$

From (3.55) and (3.56) two new limiters are defined: $\lambda_f^+$ and $\lambda_f^-$. Since the computation of the low order solution is time consuming, it is substituted in (3.55) and (3.56) to obtain:

$$\frac{\alpha_{P_{max}} - \alpha_{i_P}^{n}}{\Delta t} V_P + \sum_f F_{Cf}^{n,L} = -\sum_f \lambda_f^- A_f^{n,-} \tag{3.57}$$

$$\frac{\alpha_{P_{min}} - \alpha_{i_P}^{n}}{\Delta t} V_P + \sum_f F_{Cf}^{n,L} = -\sum_f \lambda_f^+ A_f^{n,+} \tag{3.58}$$

This solution brings, however, to low values of antidiffusive limiters [52]. This is due to the fact that stabilization of antidiffusive flux wasn't taken into account in previous assumptions. It was found out that adding on the left hand side respectively to equation (3.57) and (3.58) the opposite antidiffusive flux would lead to a higher value of the limiter while still relying on bounded physical hypothesis:

$$\frac{\alpha_{P_{max}} - \alpha_{i_P}^{n}}{\Delta t} V_P + \sum_f F_{Cf}^{n,L} + \sum_f \lambda_f^+ A_f^{n,+} = -\sum_f \lambda_f^- A_f^{n,-} \tag{3.59}$$

$$\frac{\alpha_{P_{min}} - \alpha_{i_P}^{n}}{\Delta t} V_P + \sum_f F_{Cf}^{n,L} + \sum_f \lambda_f^- A_f^{n,-} = -\sum_f \lambda_f^+ A_f^{n,+} \tag{3.60}$$

Unfortunately the solution of the previous system is undetermined because we are solving $2 \cdot N$ equations, with $N$ representing the number of control volumes while there are $2 \cdot F$ values of limiters to be determined ($F$ represents the number of faces). Therefore some simplifications must be introduced in order to find a solution to the system:

- The value of both limiters at the cell is defined ($\lambda_P^-$and$\lambda_P^+$), these are utilized on the r.h.s of equations (3.59) and (3.60)

- Limiters at the faces are computed as a combination of the limiters at the cell center of the cells that own the face as:

$$\lambda_f^+ = min(\lambda_P^+, \lambda_{NB}^-) \tag{3.61}$$
$$\lambda_f^- = min(\lambda_P^-, \lambda_{NB}^+) \tag{3.62}$$

  As a matter of fact, the *"min"* operator is used to ensure boundedness for all faces and cells. Moreover, the need to couple positive and negative limiter from respectively owner and neighbour cells arises from the fact that antidiffusive fluxes for a face have opposite signs depending on whether the cell considered is the owner or the neighbour.

- An iterative method is used to reach convergence between the value of the limiter at the cells and at the faces.

  At first iteration all limiters are set to unity ($\lambda_P^{\pm} = 1$), then the modified (3.59) and (3.60) with the first simplifications above are solved to determine the positive and negative limiters for antidiffusive fluxes at the cell ($\lambda_P^{\pm,b+1}$). And finally the face limiters are calculated from (3.61) and (3.62) as explained in the second simplification.

- All limiters are again bounded to the $[0, 1]$ interval.

In case a solution exists with a limiter greater than one, the complete antidiffusive flux prediction can be used without breaking the boundedness constraint. On the other hand, if a limiter predicted value is lower than zero there is a problem with the iterative process since it will imply that low order convective fluxes are inducing unboundedness, which is mathematically impossible. It is also important to note that the phase fraction equation will be solved iteratively and therefore, the values of low and high order fluxes can be computed with the phase fraction field of the previous iteration i.e. explicitly.

# 6 Discretised Momentum Equation

The next equation solved in the PIMPLE loop, as it is possible to see in img. 1 and 3.2, is the momentum equation. The formulation of this transport equation follows the two-phase model and hence is analogous to the one seen in chapter 2:

**Listing 3.9: Momentum equation for two-phase systems in OpenFOAM as formulated in file `UEqns.H` located in ../multiphase/reactingEulerFoam/reactingTwoPhaseEulerFoam/pU**

```
autoPtr<phaseSystem::momentumTransferTable>
    momentumTransferPtr(fluid.momentumTransfer());

phaseSystem::momentumTransferTable
    momentumTransfer(momentumTransferPtr());

{
    U1Eqn =
    (
        phase1.UEqn()
    ==
        *momentumTransfer[phase1.name()]
        + fvOptions(alpha1, rho1, U1)
    );
    U1Eqn.relax();
    fvOptions.constrain(U1Eqn);
    fvOptions.correct(U1);
}
```

The l.h.s. of the momentum equation is templated into the *basePhaseModel* and that is the discretization of the homogeneous momentum equation.

**Listing 3.10: The discretized momentum equation as implemented in the basePhaseModel**

```
tmp<fvVectorMatrix> tUEqn
(
    fvm::ddt(rho, U) + fvm::div(phi, U)
    + MRF.DDt(rho, U)
    + turbulence->divDevRhoReff(U)
==
    fvOptions(rho, U)
 );
 fvVectorMatrix& UEqn = tUEqn.ref();

 UEqn.relax();

 fvOptions.constrain(UEqn);
```

The r.h.s. consists in the *momentumTransfer*, this is a hash pointer table populated by the various models described in chapter 2 section 5. As it will be explained later in section (6.1) the pressure gradient and gravitational term won't be solved in the momentum equation but in the pressure equation to grant the P-U coupling and hence reach a convergent solution for both pressure and velocity. It is important to remember that, since we are talking

about a two-fluid system the solver has to transport two velocities and, hence the momentum equation as described in this section has to be written for both phases.

## 6.1 P-U Coupling

In CFD the velocity field is not known and emerges as part of the overall solution process along with all other flow variables [16, 17]. The solution of the momentum equation presents several problems:

- The convective terms of the momentum equation contain non linear quantities i.e. $\nabla \cdot (\rho_1 \varphi \vec{U}_\varphi \vec{U}_\varphi)$.
- The three equations that compose the momentum (x-direction, y-direction, z-direction) are strictly linked .
- The pressure gradient play a difficult role since no equation for it is present.

A solution can be found coupling pressure and velocity and hence introducing a constraint in the solution of the flow field: if the correct pressure field is applied in the momentum equations the resulting velocity field should satisfy continuity [16, 17]. Apart from the need of a p-U coupling algorithm, that in our case is represented by the PIMPLE loop, another feature is implemented and important to highlight: the correction of the velocity field. As seen in section (4.3) Rhie and Chow [51] defined a procedure for which a velocity at the center cell is defined as the interpolated value of the velocities at the faces (predicted value) plus a correction term. This term can be different from author to author but the more physical it is, the faster the solution will converge. In our case, the pressure equation, and hence the p-U coupling presents several terms for the correction. The corrected flux can be written as:

$$\begin{aligned}\phi_\varphi^{**} =& \phi_\varphi^* - \frac{D^{TD} + p'}{\left(\mathcal{A}_\varphi\right)_D} S_f \nabla \alpha_\varphi - \frac{F^{L+WL}}{\left(\mathcal{A}_\varphi\right)_D} S_f - \frac{K_D}{\left(\mathcal{A}_\varphi\right)_D} \phi_{1-\varphi} \\ & - \frac{\alpha_\varphi}{\left(\mathcal{A}_\varphi\right)_D} \Big[ g h_f S_f \nabla \rho_\varphi - \alpha_{1-\varphi} (\rho_1 - \rho_2)_f (g \cdot S_f) \Big]\end{aligned} \tag{3.63}$$

Where:

$\phi_\varphi^*$: The predicted flux;

$D^{TD}$: Turbulent diffusivity coefficient;

$p'$: Pressure derivative with respect to volume fraction ;

$F^{L+WL}$: Force due to lift and wall lubrication force;

$K_D$: Drag coefficient;

The last term accounts for compressibility and gravitational effects.

# 7 Discretised Energy Equation

Analogously to what said in the previous paragraph, the energy equation is discretized and implemented in OpenFOAM as follows:

**Listing 3.11: Energy equation for two-phase systems as formulated in file `EEqns.H` located in: application/solvers/multiphase/reactingEulerFoam/reactingTwoPhaseEulerFoam**

```
autoPtr<phaseSystem::heatTransferTable>
        heatTransferPtr(fluid.heatTransfer());

    phaseSystem::heatTransferTable&
        heatTransfer = heatTransferPtr();

    {
        tmp<fvScalarMatrix> E1Eqn(phase1.heEqn());

        if (E1Eqn.valid())
        {
            E1Eqn =
            (
                E1Eqn
             ==
               *heatTransfer[phase1.name()]
              + alpha1*rho1*(U1&g)
              + fvOptions(alpha1, rho1, phase1.thermo().he
                ())
            );

            E1Eqn->relax();
            fvOptions.constrain(E1Eqn.ref());
            E1Eqn->solve();
        }
    }
```

As already said the l.h.s. of the equation is templated and can be recovered in the *basePhaseModel* as the discretized implementation of the energy equation:

**Listing 3.12: Energy equation for a two-phase system in OpenFOAM.**

```
fvScalarMatrix E1Eqn
     (
         fvm::ddt(alpha, rho, he) + fvm::div(alphaRhoPhi,
            he)
       - fvm::Sp(contErr, he)

       + fvc::ddt(alpha, rho, K) + fvc::div(alphaRhoPhi, K)
       - contErr*K
       + (
             he.name() == thermo.phasePropertyName(e)
           ? fvc::div(fvc::absolute(alphaPhi, alpha, U), p)
           + p*fvc::ddt(alpha)
           : -alpha*dpdt
         )

       - fvm::laplacian
         (
             fvc::interpolate(alpha)
            *fvc::interpolate(thermo.alphaEff(phase.
               turbulence().mut())),
             he
         )
     );

     E1Eqn.relax();
```

Continuing the analogy with the previous section, the heat transfer table is populated with the heat transfer model coefficients that take into account the heat transfer between phases. It is also important to remember that the phase change introduces an heat source/sink term due to the latent heat of evaporation, as illustrated in section 2.2.2.

# Chapter 4

# Simulations

In this chapter two benchmark simulations, described in [13], will be performed on two OpenFOAM solvers, namely `twoPhaseEulerFoam` and `interFoam`. The objective is to highlight the differences between Eulerian-Eulerian and VoF approach as implemented in OpenFOAM, in particular the benchmark aims to highlight how sharply a method can transport the interface of a bubble.

## 1 Introduction to the Case-study

The two case-studies presented in [13] will be performed alternatively on the two solvers, the results will firstly be discussed in terms of convergence rate towards the finest solution as well as accuracy of bubble interface tracking comparing the gaseous bubble interface ($\alpha_{gas} = 0.5$) with the literature both theoretically 2.1 and experimentally/numerically [13]. As will be thoroughly described before in the chapter the two cases refer to two different bubble regimes. The first setup is characteristic of ellipsoidal cap bubbles while the second one is typical of skirted bubbles. These two cases were designed to spot two different aspects of interface tracking quality:

- **Case 1:** Allows to detect how sharply the interface is tracked;
- **Case 2:** Allows to detect whether or not the solver is capable to track breakup and how precisely the breakup bubbles are described.

The VoF solver `interFoam` was taken as reference since, as mentioned before in the thesis, VoF has been widely used for interface tracking dominating problems and has been proven to provide satisfying results in interface tracking. Not to leave anything to the chance the cases were ran before

through the VoF methodology, after that the same cases were ran using `twoPhaseEulerFoam` and the results were then compared. All the simulations were performed in order to evaluate two-fluid method's capacity of interface tracking and to provide some means that will help for the identification of eventual problems.

## 2 Cases Setup

The initial configuration is identical for both cases and consists of a circular bubble of diameter $d_0 = 0.5$ centred at $[0.5; 0.5]$ in a $[1 \times 2]$ rectangular domain, as represented in figure 4.1. The density of the bubble is lower than that of the surrounding fluid ($\rho_b < \rho_l$). No-slip boundary conditions are applied on top and bottom walls ($\vec{u} = \vec{v} = 0$), while the free slip condition is applied to the vertical walls ($\vec{u} = 0$).

Table 4.1 lists the fluids and physical parameters which specify the test cases. The evolution of the bubbles should be tracked for 3 time units during which the defined benchmark quantities should be measured. As described in section 2.1 of chapter 1 adimensionalization is frequently used to assist with classifying simulations and can be introduced by scaling through a characteristic length and time:

$$L = d_0$$

$$t = \frac{L}{U_g}$$

where $U_g = \sqrt{g d_0}$ is the gravitational velocity.

The first test case models a rising bubble with $\mathcal{R}e = 35$, $\mathcal{E}o = 10$ and both density and viscosity ratio set to 10. According to experimental studies [28] as highlighted in figure 1.8 (chapter 1), such a bubble will end up in ellipsoidal regime. Extending the validity also to 2D flows it would mean that the surface tension effects are strong enough to hold up the bubble and hence no breakup is expected in this test case.

The second models a rising bubble with $\mathcal{R}e = 35$, $\mathcal{E}o = 125$ and large values of density and viscosity ratios, respectively 1000 and 100. This bubble lies somewhere between the skirted and dimpled ellipsoidal-cap regimes indicating that breakup can possibly occur [28] [Fig. 1.8], which will present additional challenges to the two solvers in interface tracking.

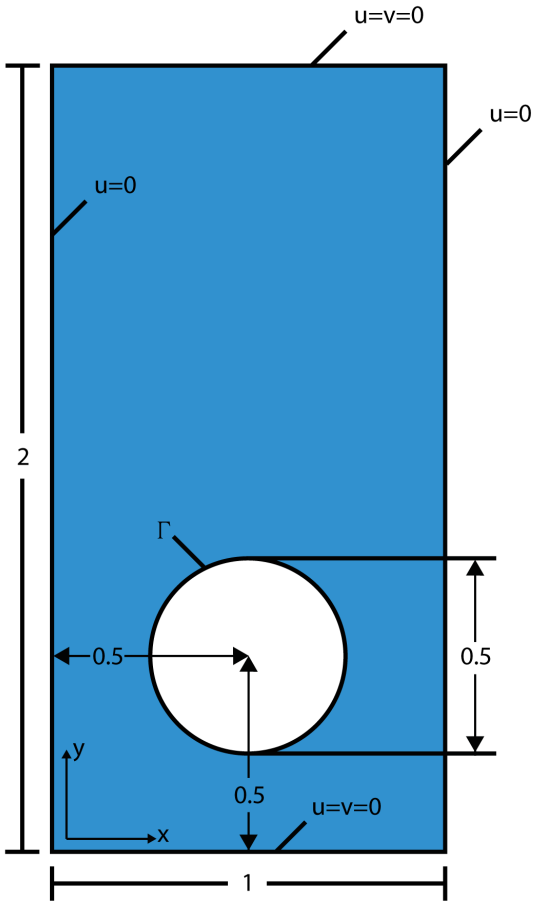


Figure 4.1: Initial configuration and boundary conditions for the test cases.

Table 4.1: Pysical parameters defining the test cases

| Test Case | $\rho_l$ | $\rho_g$ | $\mu_l$ | $\mu_g$ | g | $\sigma$ | $\mathcal{R}e$ | $\mathcal{E}o$ | $\rho_l/\rho_g$ | $\mu_l/\mu_g$ |
|---|---|---|---|---|---|---|---|---|---|---|
| 1 | 1000 | 100 | 10 | 1 | 0.98 | 24.5 | 35 | 10 | 10 | 10 |
| 2 | 1000 | 1 | 10 | 0.1 | 0.98 | 1.96 | 35 | 125 | 1000 | 100 |

## 2.1 Benchmark Quantities

Visual comparison of the interface transport against time is the most obvious way to compare two simulations. However, this does not allow to accurately determine the resolution of a method. Therefore the following benchmark quantities are introduced in order to provide the reader with a quantitative numerical confrontation method regarding both interface transport accuracy and computational effort:

***Point quantities:*** Positions of various points can be used to track the translation of bubbles. It is common to use the centroid (center of mass) defined by:

$$\mathbf{X_c} = (x_c, y_c) = \frac{\int_{\Omega_g} x dx}{\int_{\Omega_g} 1 dx} \tag{4.1}$$

where $\Omega_g$ denotes the region that the gaseous bubble occupies.

***Circularity:*** The degree of circularity introduced by [53], can be defined as:

$$c = \frac{P_a}{P_b} = \frac{\text{perimeter of area-equaivalent circle}}{\text{perimeter of bubble}} = \frac{\pi d_a}{P_b} \tag{4.2}$$

***Rise velocity:*** The mean velocity with which the bubble is rising or moving is an interesting quantity, since it does not only measure how the interface tracking is working but also the quality of the overall solution, it is defined as:

$$\mathbf{U_c} = \frac{\int_{\Omega_g} \vec{u} dx}{\int_{\Omega_g} 1 dx} \tag{4.3}$$

where $\Omega_g$ again denotes the region that the bubble occupies. A possible variant is to use the velocity at the centroid of the bubble. Important to mention is that the rise velocity (opposed to the gravitational vector) will reach a stationary limit called terminal velocity.

## 2.2 Error quantification

The temporal evolution of the computed benchmark quantities can be measured against suitable reference solutions to establish the following relative error norms:

$$l_1\text{error:} \qquad ||e||_1 = \frac{\sum_{t=1}^{NTS} |q_{t,ref} - q_t|}{\sum_{t=1}^{NTS} |q_{t,ref}|} \tag{4.4}$$

$$l_2\text{error:} \qquad ||e||_2 = \left( \frac{\sum_{t=1}^{NTS} |q_{t,ref} - q_t|^2}{\sum_{t=1}^{NTS} |q_{t,ref}|^2} \right)^2 \tag{4.5}$$

$$l_\infty\text{error:} \qquad ||e||_\infty = \frac{max_t |q_{t,ref} - q_t|}{max_t |q_{t,ref}|} \tag{4.6}$$

where $q_t$ represent the temporal evolution of quantity q. The solution computed on the finest grid with the smallest time step is usually taken as a

reference solution $q_{t,ref}$. With the relative errors established and CPU times measured, it is then straightforward to see how much effort is required to establish a certain accuracy. Additionally, convergence rates (ROC) for the quantities can also be computed as:

$$ROC = \frac{log_{10}\Big(||e^{l-1}||/||e^{l}||\Big)}{log_{10}\Big(h^{l-1}/h^{l}\Big)} \tag{4.7}$$

where $l$ is the grid refinement level and $h$ the mean cell edge length.
The ROC will not indicate how well a method converges to the exact solution, but how it converges toward an approximate discrete solution. Although this means that absolute convergence cannot be proved, if the results from a sufficient number of methods approach the same solution it can be assumed that the difference between the discrete reference and the exact solution is not too large.

In this work, contrarily to what was done in [13], a first analysis will be performed on case-study 2, in order to understand how the two methods can track phase interface in breakup conditions. Following in the chapter also the case-study 1 will be performed and analysed in order to highlight how surface tension forces are accounted by the two methodologies.

# 3 Case 2 Results

## 3.1 `interFoam`

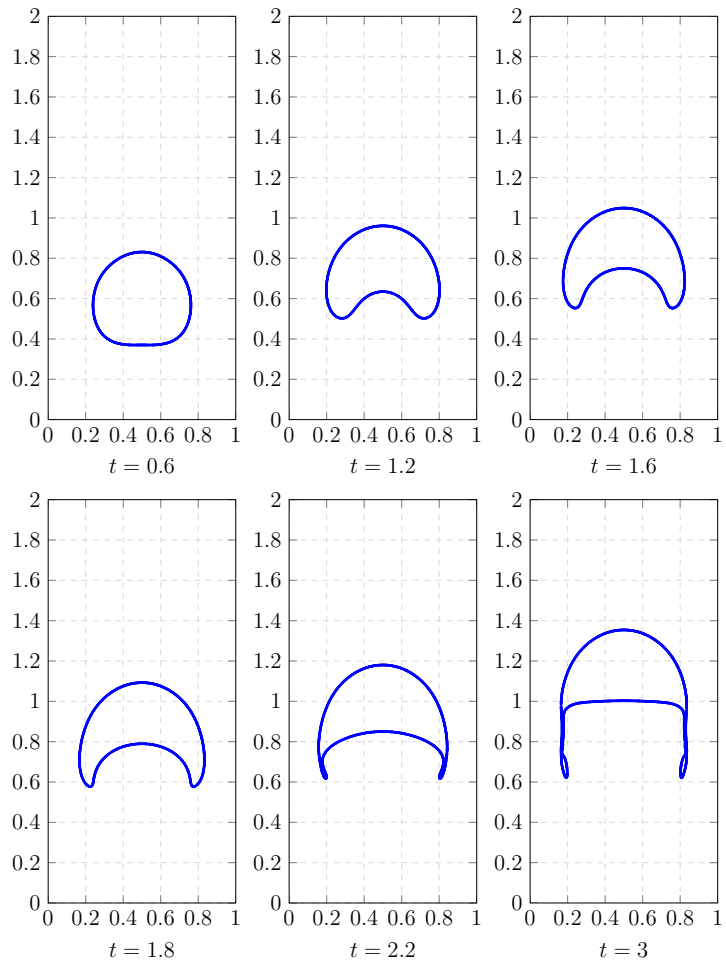


Figure 4.2: Contour plot for $h = 1/640$.

Following the solutions computed through the VoF solver `interFoam` will be presented. In figure 4.2 is presented the bubble evolution over time for the finest mesh resolution. The bubble contours were considered at $\alpha_{gas} = 0.5$ and the progressive refinements were defined through rectangular tensor product grid with cell size $h = 1/[40, 80, 160, 320, 640]$. From 4.2 it is evident, recalling section 2.1 described in chapter 1, that a bubble breakup happens after $t = 2.2$ and the "breakup-tails" are transported consistently, with respect to both theory references [2.1] and the benchmark tests [13].

Figure 4.3 presents a focus on how breakup is captured by the method at different mesh resolution, in order to do so the gas phase contours at time $t = 3$ were compared highlighting how mesh resolution strongly influences how "breakup-tails" are captured or not and also how the droplets that compose the tails assume different shapes at different resolutions. It is evident that the solution on the coarsest grid produces a rather unphysical interface with sharp-edged tails, on the other hand it seems that the bulk bubble is captured correctly with respect to the solution on the finest grids.

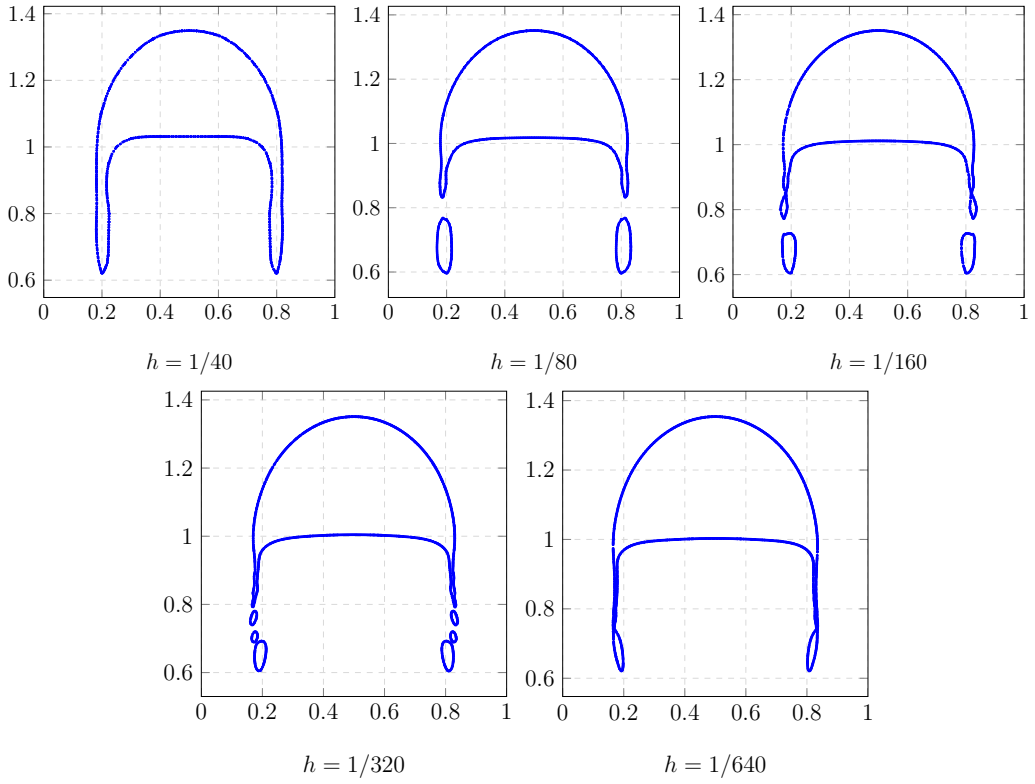


Figure 4.3: Contour plot at time $t = 3$ for different mesh refinements $h$.

The circularity, represented in figure 4.4, is constant until around $t = 0.5$ and then decreases almost linearly. It is important to highlight that around $t = 2.4 \div 2.8$ on the finest grids a change of concavity in bubble-circularity chart is present, this is due to the bubble breakup. Since the coarsest case ($h = 1/40$) cannot track bubble breakup, this change of concavity is not captured.

The proof of that lays in figure 4.4 (b), where no bubble breakup is tracked. On the other hand starting from $h = 1/80$ an incipient breakup can

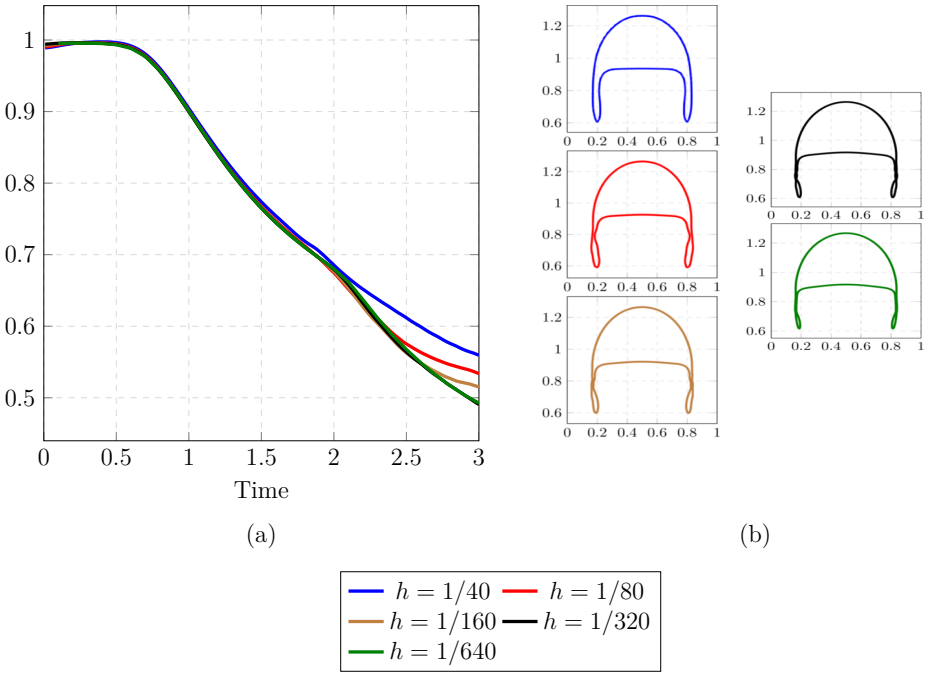


Figure 4.4: Circularity at different mesh resolutions for the VoF methodology (a), focus on bubble contours at incipient bubble breakup ($t = 2.6$)(b).

be identified.

Time evolutions of the center of mass and mean rise velocity of the bubble are shown respectively in figure 4.5 (a) and (b). The center of mass moves almost linearly and only slight deviations can be spotted in low mesh resolution simulations. For the finest solutions the trend is similar and so the lines almost superpose.

This, contrarily to what said for bubble circularity, doesn't allow to locate bubble breakup. Moving to the mean rise velocity, it almost superpose in the first time unit of simulation for all mesh resolutions, then it is possible to identify a gap between different grid refinements. Namely, the two local maxima are quite different (see table 4.2), but it is possible to spot both from figure 4.5 (b) and from table 4.2 that convergence toward the finest grid is reached. Also bubble mean rise velocity doesn't allow to locate bubble breakup and even tough a solution gap is identified in correspondence to the local maxima, the maxima themselves are identified at all mesh resolution within a reasonable distance.

The errors and rate of convergence will not be calculated for this case study to be consistent with the reference benchmark paper [13]. This is due

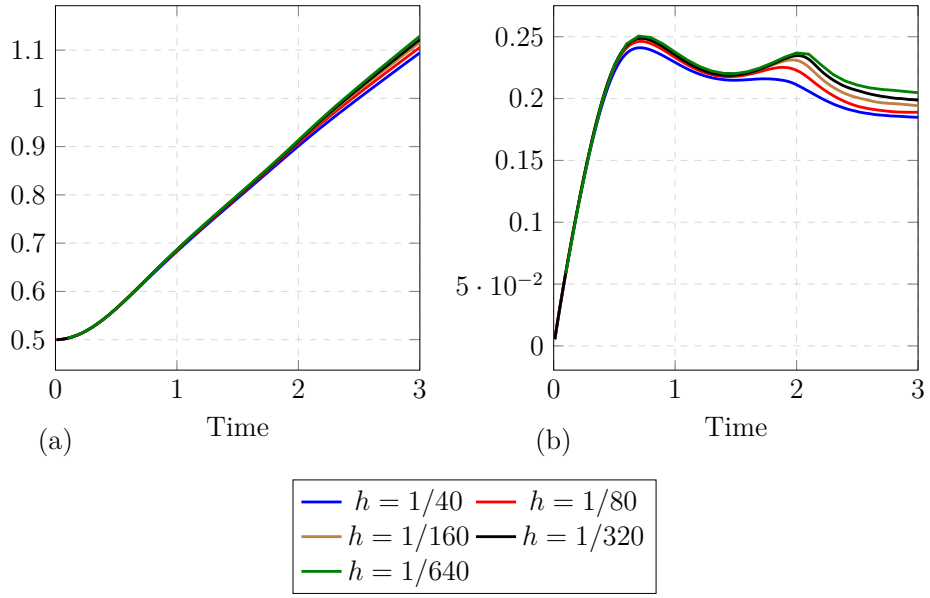


Figure 4.5: Centroid position over time (a), mean rise velocity (b) calculated for VoF method at different mesh resolutions.

Table 4.2: Minimum circularity and maximum rise velocities with corresponding time and centroid final position

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.5594 | 0.53374 | 0.51518 | 0.49057 | 0.49252 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max,1}$ | 0.2411 | 0.24622 | 0.24876 | 0.2487 | 0.2506 |
| $t\|_{v=v_{max,1}}$ | 0.70937 | 0.72031 | 0.71875 | 0.72031 | 0.7210 |
| $v_{max,2}$ | 0.21598 | 0.22525 | 0.23125 | 0.23474 | 0.23694 |
| $t\|_{v=v_{max,2}}$ | 1.7406 | 1.8797 | 1.9688 | 2.0094 | 2.0105 |
| $y_c\|_{t=3}$ | 1.0945 | 1.1059 | 1.1161 | 1.1219 | 1.1291 |

to the fact that the solutions are visibly different and convergence towards the finest solution can be appreciated graphically, the computation of ROC will be required during case 1 discussion, where the diverse solutions will appear as a superposed bubbles. In that case the error computation will help the reader understand whether the solution is converging towards the finest grid one or not.

The findings achieved through this simulation are in line with literature and the benchmark paper results. This means that, as said before in the chapter, The VoF methodology is able to capture bubble breakup correctly

almost at any rate of refinement. Obviously the more refined a simulation is the more accurate the solution will be, in terms of both bubble velocity and shape/breakup of the interface. It is straightforward to understand that mesh resolution is the limiting variable for breakup capturing, i.e. if mesh dimensions are bigger or in the order of breakup-bubble dimensions, no breakup will be captured. On the other hand too refined solutions will provide an accurate solution but uselessly long to simulate. The right mesh-element sizing has to be chosen in order to provide a correct interface tracking while containing the computational time.

## 3.2 `twoPhaseEulerFoam`

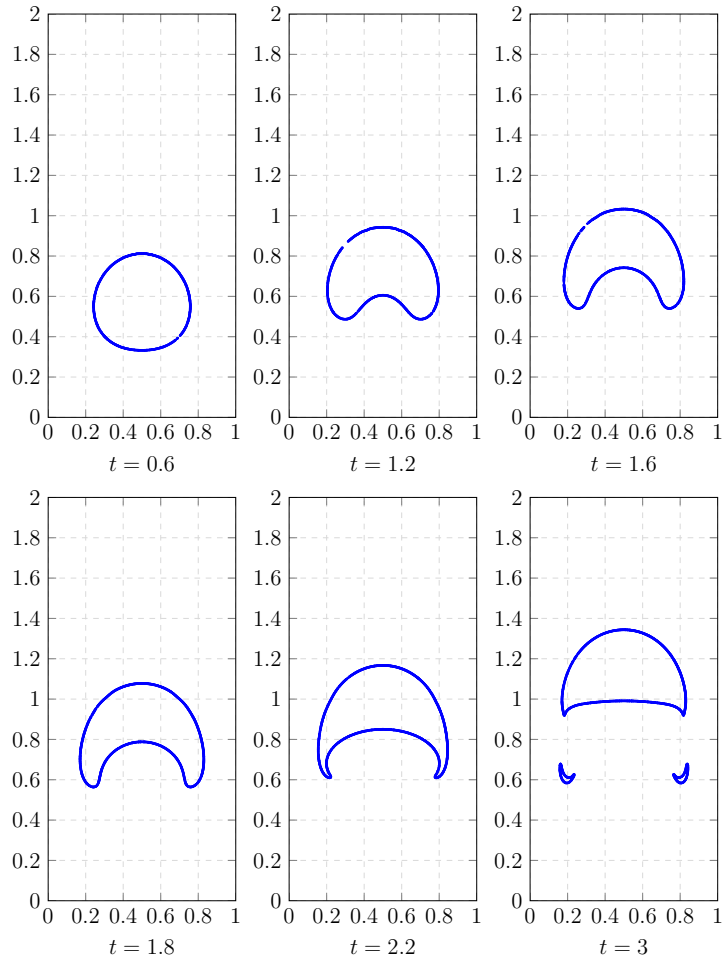


Figure 4.6: Contour plot for $h = 1/640$.

Following the solutions computed through the two fluid solver `twoPhaseEuler Foam` will be presented. In figure 4.6 is reported the bubble evolution over time at $h = 1/640$ mesh refinement. For this case study the bubble contours were considered at $\alpha_{gas} = 0.5$ and the progressive refinements were defined through rectangular tensor product grid with cell size progressively defined as : $h = 1/[40, 80, 160, 320, 640]$. From 4.6 it is evident, recalling section 2.1 described in chapter 1, that a bubble breakup happens after $t = 2.2$ and the "breakup-tails" are not tracked correctly by the model, with respect to both theory references [2.1] and the benchmark test [13]. Figure 4.7 presents a focus on how breakup is captured by the method at different mesh resolu-

tion, in order to do so the gas phase contours at time $t = 3$ were compared highlighting how mesh resolution strongly influences how "breakup-tails" are captured. In this case it is possible to conclude that trailing breakup bubbles are not tracked before $h = 1/320$ and even at this resolution only the biggest core-bubbles of the tails are detected by the solver, giving no idea on how the bubble interface is transported. This highlights how mesh demanding the solver is and probably that some interface capturing problems are present. It is evident that the solution on the coarsest grid produces a rather unphysical contour, not only the tails are not captured (as well as for $h = 1/80$,$h = 1/160$) but also the core bubble shape and position is not tracked correctly. However convergence towards the finest solution may seem to be reached as it is possible to prove from table 4.3.

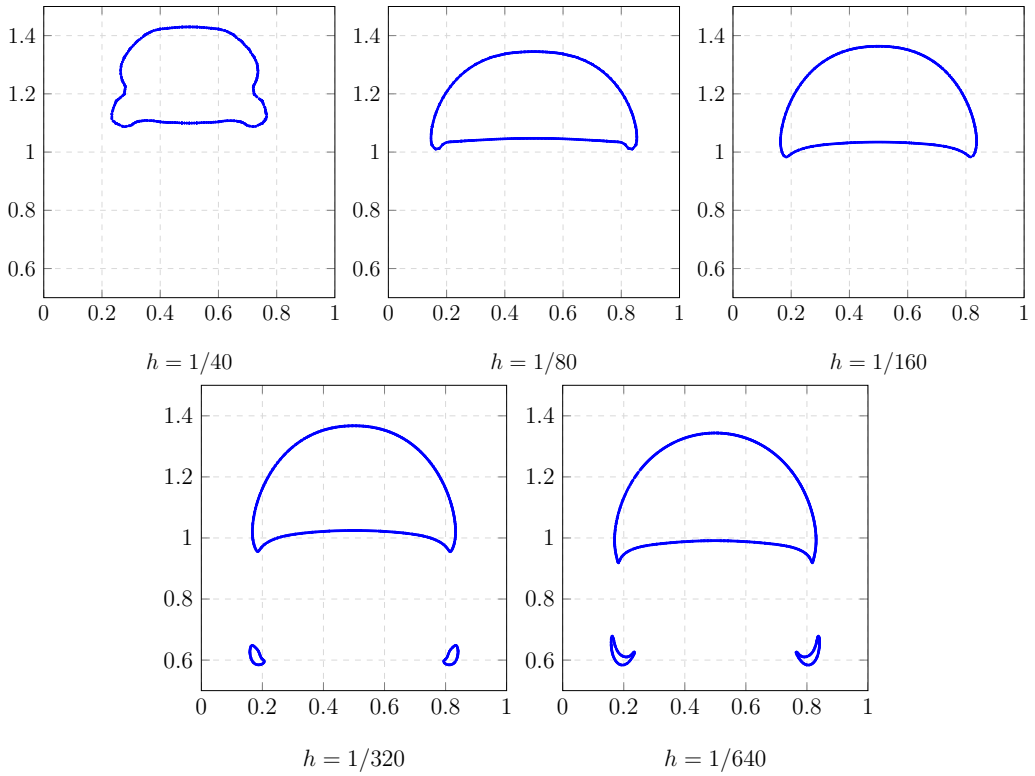


Figure 4.7: Contour plot at time $t = 3$ for different mesh refinements $h$.

The circularity, represented in figure 4.8, is constant until around $t = 0.5$ and then decreases almost linearly. It is important to highlight that around $t = 2 \div 3$ on the finest grids a change of concavity in bubble-circularity chart is present, this is due to the bubble breakup. The coarsest case ($h = 1/40$ and

$h = 1/80$) present the same trend earlier in time with respect to the finest solutions and sue to the diffusivity of the solution no breakup can be tracked in the latest time-steps. Figure 4.9 shows how the bubble breakup causes the change of concavity in the bubble-circularity chart and how the breakup correspond exactly to the local maximum of the curve. Similar behaviour can be spotted in 4.8 for the finest mesh (green line) but the breakup is translated in time as the inflection point position may suggest. This is also the proof that for the brown contour ($h = 1/160$) breakup happens at $t = 2.2$ but the "tails" do not appear in the last time-step.

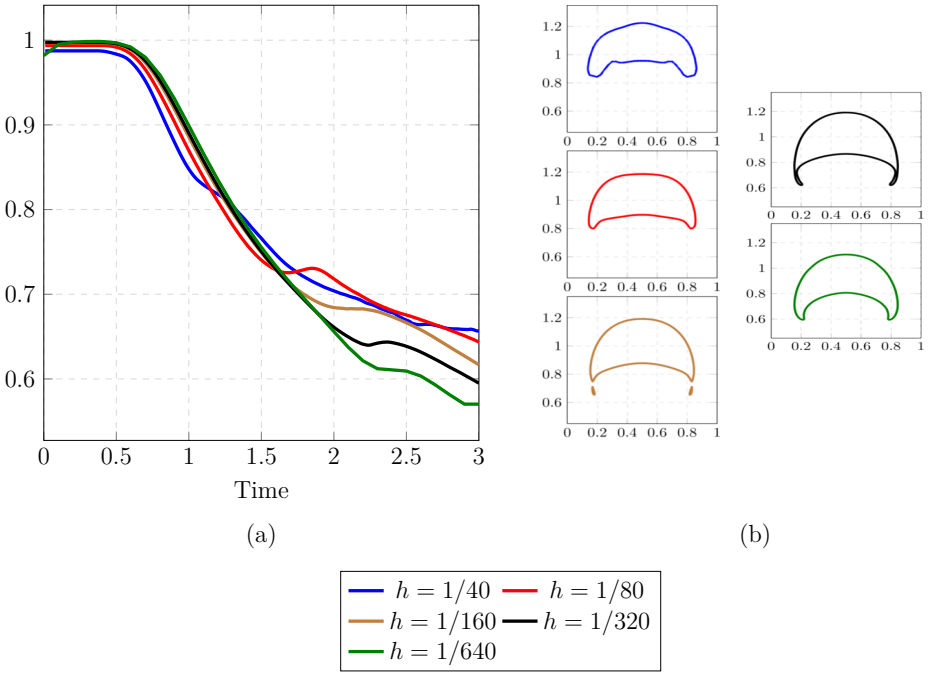


Figure 4.8: Circularity at different mesh resolutions for the two fluid methodology (a), focus on bubble contours at incipient bubble breakup ($t = 2.2$)(b).

Time evolutions of the center of mass and mean rise velocity of the bubble are shown respectively in figure 4.10 (a) and (b). The center of mass moves almost linearly and only slight deviations can be spotted at low mesh resolution simulations. For the finest solutions the trend is similar and so the curves almost superpose.

This, contrarily to what said for bubble circularity doesn't allow to locate bubble breakup. Moving to the mean rise velocity, it almost superpose in the first time unit of simulation for finest mesh, but the coarsest show an unphysical top raise velocity and then a local minimum way too low with

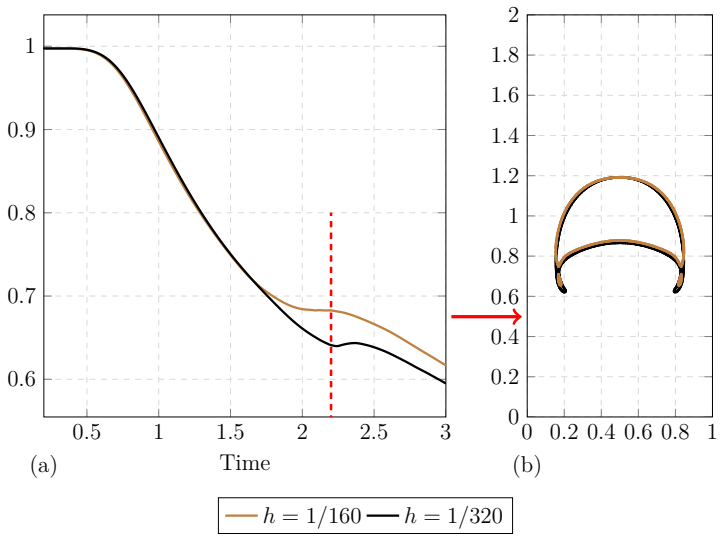


Figure 4.9: Bubble circularity evolution against time (a) for different mesh resolutions and bubble contours at $t = 2.2$ for two fine resolutions (b)

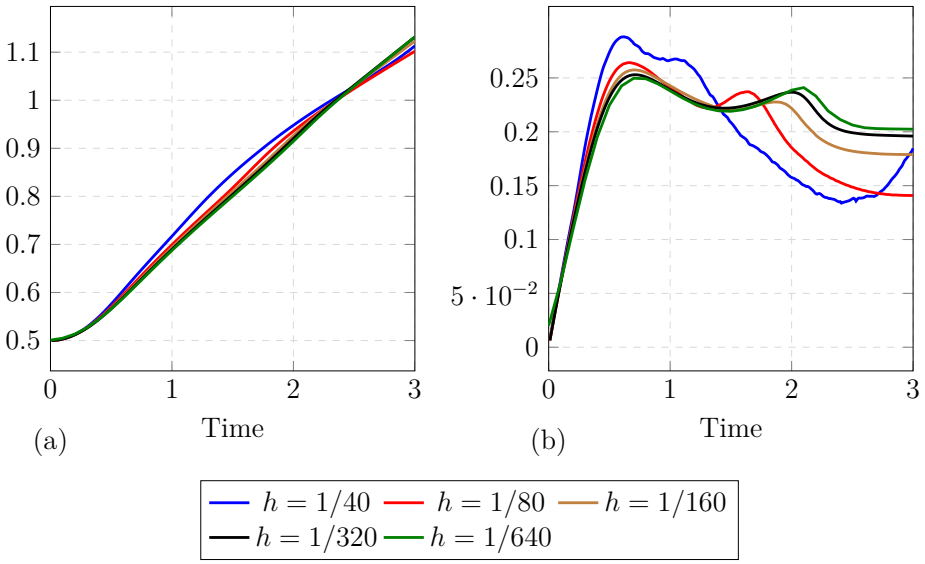


Figure 4.10: Centroid position over time (a), mean rise velocity (b) calculated for two fluid method at different mesh resolutions.

respect to the other simulations. This, added to the other considerations made in the section and to the results presented in table 4.3, lead to the conclusion that being the solver really mesh demanding the solution at $h =$

1/40 is unpysical and can't produce a satisfying solution. $h = 1/80$ shows a more converging trend, the maximum is aligned to the finest ones in the first time-steps but then the second is really premature and follows a unphysical velocity reduction. The two local maxima are quite different (see table 4.3), so it is possible to say both from figure 4.10 (b) and from table 4.3 that convergence toward the finest grid is not reached. The inequalities found in the mean rise velocity chart are surely due to the lack of resolution given by coarsest grids and as a consequence to the incorrect interface tracking that derives from that. This can lead to the conclusion, even without checking the errors that the solutions on the coarsest grids cannot be considered sufficient because neither interface nor bubble rise velocity are traced correctly.

Table 4.3: Minimum circularity and maximum rise velocities with corresponding time and centroid final position

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.65637 | 0.64369 | 0.6167 | 0.595 | 0.5702 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max,1}$ | 0.28816 | 0.26418 | 0.25739 | 0.25295 | 0.24972 |
| $t\|_{v=v_{max,1}}$ | 0.62 | 0.66 | 0.7 | 0.71 | 0.7 |
| $v_{max,2}$ | 026759 | 0.23716 | 0.22766 | 0.23672 | 0.24102 |
| $t\|_{v=v_{max,2}}$ | 1.04 | 1.64 | 1.85 | 2 | 2.1 |
| $y_c\|_{t=3}$ | 1.1131 | 1.102 | 1.1231 | 1.1317 | 1.1320 |

For what concerns the finest meshes simulations, it is evident that convergence is reached towards the finest solution, which provides a coherent rise velocity and bubble centroid position but it is not able to track correctly the breakup of the bubble. This will be discussed more thoroughly in the following paragraph where a comparison between the two solvers will be carried on.

## 3.3 Comparison

Figure 4.11 reports a comparison between the two solvers on how the interface is transported, at the last time-step, for the same mesh resolution. Starting from top left side of the picture it is evident that the coarsest mesh gives an unphysical solution for both solvers. However the VoF solver (in red) *interFoam* present a sharper bubble interface and it is also able to capture the "tails" of the bubble, the centroid position and so the bubble velocity as well as its size are similar to the ones derived for the finest meshes (4.11) and in the reference paper [13]. Moving the analysis to a finer ($h = 1/80$)

mesh it is possible to identify a bubble breakup, this highlights that the solution is more consistent to the finest grids and so represent a more physical simulation. The finest meshes are able to capture the single breakup-bubbles composing the tails. On the other hand (in blue) we have the simulations computed through the two-fluid solver *twoPhaseEulerFoam*. The coarsest mesh solution is unpysical from both shape and position point of view, the interface is irregular and no tail is captured, moreover the bubble centroid reaches a visibly higher position with respect to VoF solution as well as finer mesh solutions. Bubble dimensions are significantly lower than the VoF ones, this suggests that diffusion phenomena are acting on the interface. These

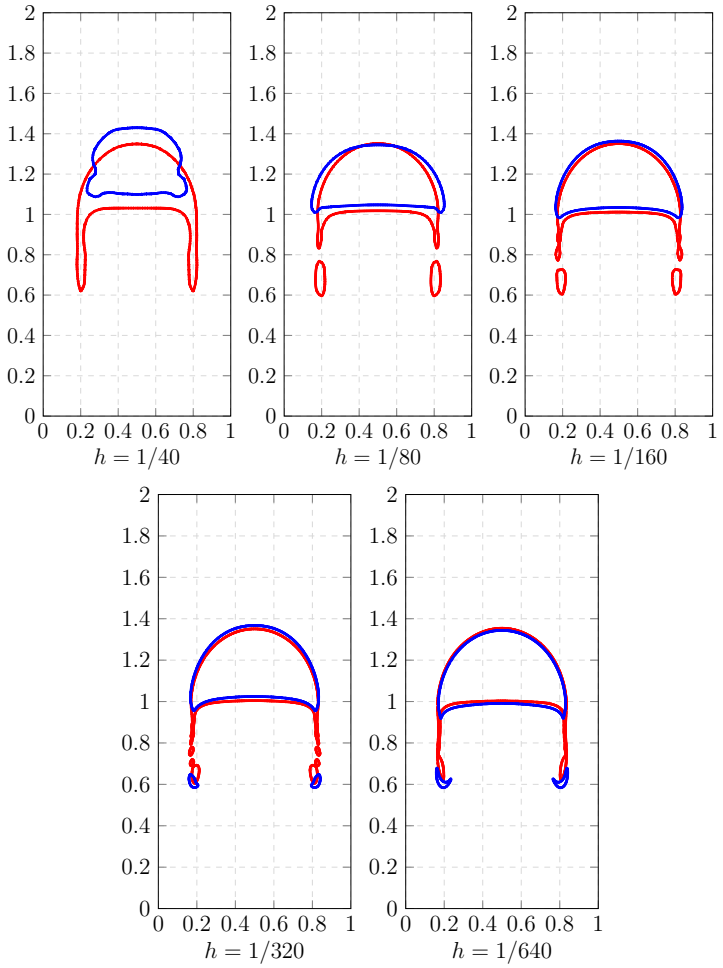


Figure 4.11: Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions.

bad results are surely, at least in part, due to the mesh resolution. Moving to finer simulations the results improve, even though until $h = 1/320$ no breakup can be identified by the solver. It is interesting to notice how with mesh refinement the core bubble tend to superpose to the (correct) VoF solution. This means that both mean bubble velocity and centroid position over time are captured correctly, but breakup is not identified except for the biggest tail-bubbles. This could be either due to diffusivity of the solver or other modelling problems of the solver. This conclusion is reinforced by what said earlier in section 3.2 regarding bubble breakup, breakup-bubbles are not tracked in coarsest meshes' solutions because breakup happens earlier in time and their interface diffuse before the last time-step.

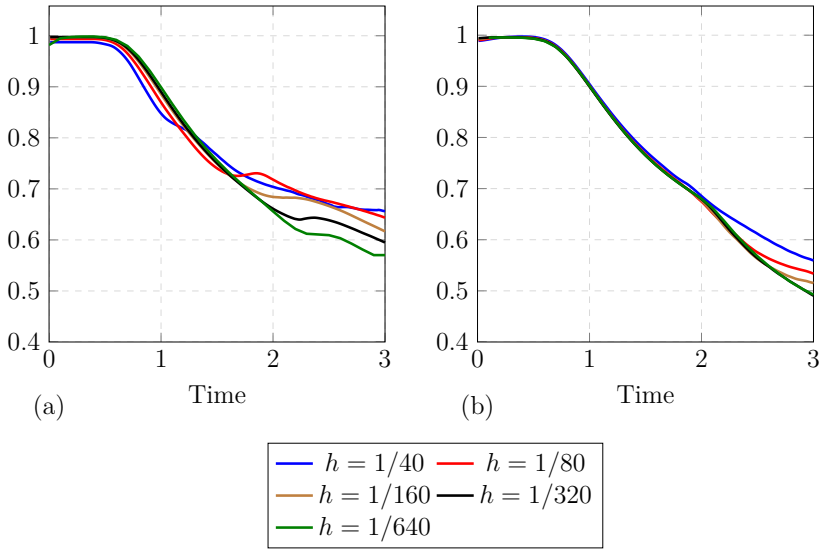


Figure 4.12: comparison between bubble circularities at different mesh resolutions for both two-fluid method (a) and VoF (b).

This highlights a twofold problematic on the one hand we have premature breakup on coarsest meshes and on the other a diffusive solver not able to track small bubbles.
This highlights how a more complex solver, that should better track the two fluids since a momentum equation is solved for each phase produces worse solutions with respect to a VoF methodology solver. On the one hand this is surely due to the mesh demand of the two-fluid solver and probably, on the other an interface-capturing problem can be present as well.

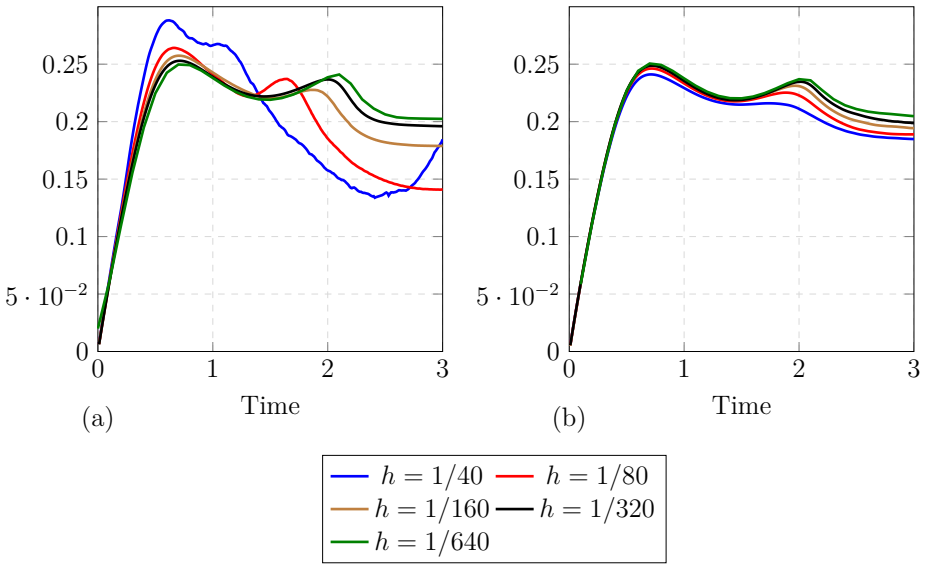


Figure 4.13: comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a) and VoF (b).

As it is possible to see from figures 4.12,4.13 and 4.14 the other benchmark quantities are not comparable except in some measure for the finest grid refinements. Starting from bubble circularity it can be said that the evolution trend is similar in the two methodologies, namely for the first 0.5 time units the circularity remains almost constant and then decreases. Two main differences can be spotted in figure 4.12, neglecting the two coarsest simulations for the two-fluid method, we have in 4.12(a) a higher final value of circularity and a change of concavity situated between 2 and 2.5. As already highlighted in the previous section the change of concavity is the result of bubble breakup that happens at significantly different time-steps in the two-fluid model. The higher final value of circularity is due to the fact that except for the core bubble of the tails no other breakup is tracked in the two fluid method, giving a solution with a more bubble-like interface. This results

are consistent with what said earlier for the bubble contours and support the theory of a problem of interface capturing in *twoPhaseEulerFoam.*
When talking about unphysicality of the coarsest solution of the two fluid method figure 4.13 gives a tangible overview of what we are talking about. It is evident that for the case $h = 1/40$ a huge overestimation error is committed in the early time-steps, while on the other hand on the last ones an opposite error of underestimation is committed, in addition to this the second local maximum results way anticipated with respect to the other solutions. Similar conclusions can be derived for the finer case $h = 1/80$ that is able to give a better solution for the earlier time-steps but presents both the advance and the underestimation problems spotted for the coarsest simulation. Increasing the mesh resolution we obtain a profile similar to the one derived through VoF simulation. This is consistent with the analysis done on the contours earlier in the section, since the main velocity is linked to the core bubble and since the two-fluid method is able to capture correctly the interface of that bubble, it is able to deliver a correct mean velocity profile even though it is not able to track the breakup.
This consideration is valid also for the bubble centroid. Being this quantity influenced mainly by the core bubble, using finer meshes we can obtain a correct tracking of the bubble centroid even without any tracking of the "breakup-tails". As reported in figure 4.14 we see that evident deviations are present for $h = 1/40$ and $h = 1/80$ but the other mesh resolution trends are in line with the VoF simulations.

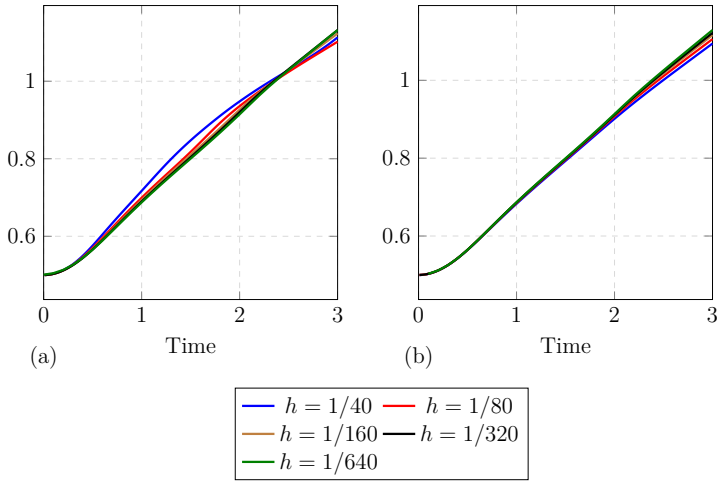


Figure 4.14: comparison between bubble centroids at different mesh resolutions for both two-fluid method (a) and VoF (b).

# 4 Case 1 Results

As mentioned before in the chapter, the first numerical benchmark case presented in [13] was tested as well. This, contrarily to the previous case, simulates a bubble in the ellipsoidal range (Chapter 1 section 2.1). The scope of this case is to highlight whether the solver is capable of capturing the interface of the bubble, since no breakup is present the benchmark quantities will track solely the quality of the interface. This second benchmark test will then give the means to determine whether the two-fluid methodology implemented in Open$\nabla$Foam through `twoPhaseEulerFoam` presents an interface-tacking problem or not. Namely it will allow to determine the causes of the differences highlighted in the previous section between two-fluid and VoF.

## 4.1 `interFoam`

First of all the benchmark case is solved with the VoF solver `interFoam`, to set a reference. This is done because VoF methodology was already proven in literature to give satisfying enough results in interface tracking as already mentioned in the introduction of this work and as resumed in [13].
Following the bubble contours at the last time-step are reported for all the consecutive mesh refinements to provide the reader with a visual confrontation.

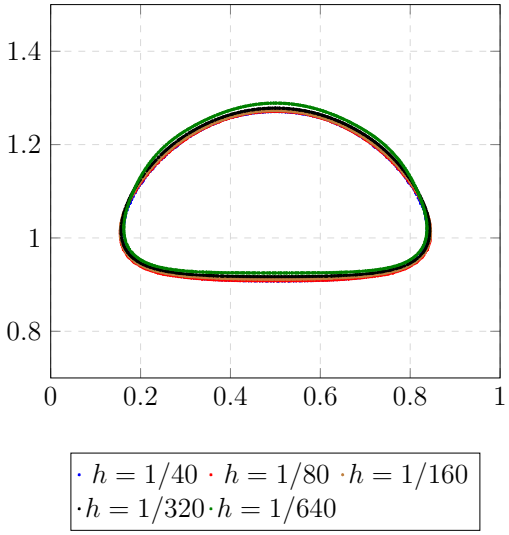


Figure 4.15: Visual confrontation of contours at last time-step for different mesh refinement .

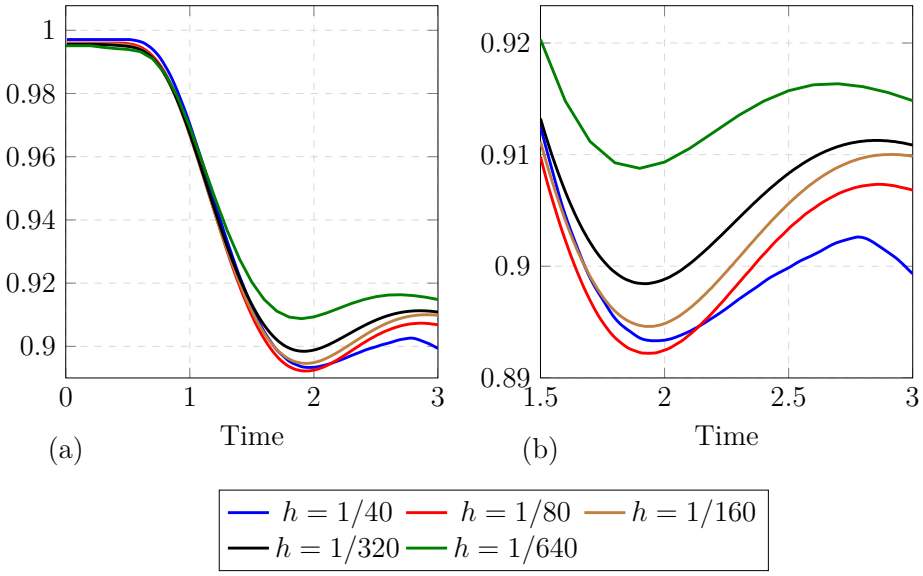


Figure 4.16: bubble circularity over time (a), zoom of bubble circularity in the second half of bubble evolution (b) calculated for VoF method at different mesh resolutions.

Figure 4.16(a) represents bubble circularity over time. As in the previous case study the first 0.5 time units present a constant value of bubble circularity, which decreases linearly until around $t = 1.5$, then a local minimum is reached around $t = 2$ and the final value of circularity ranges between $c = 0.9$ and $c = 0.91$. The excursion in bubble circularity is so small that is almost imperceptible from picture 4.15, that's why the relative errors as defined in section 2.2 were introduced. If we zoom the bubble circularity chart (Fig. 4.16(b)) it is possible to identify small variations between the different mesh refinement but the values lay in a range of $10^{-3}$. The local minima are located almost at the same time-step except for the coarsest case, which deviates from the other solutions also in the last time-steps. It is safe to say, even without consulting the errors, that global convergence towards the finest solution is reached. This conclusion is supported also by 4.17 where respectively centroid position over time (a) and the mean bubble rise velocity (b) are reported. Just minimal deviations are captured around the final time-steps but these differences only partly influence the bubble evolution, since, as it is possible to see from picture 4.15, almost all interfaces are congruent.

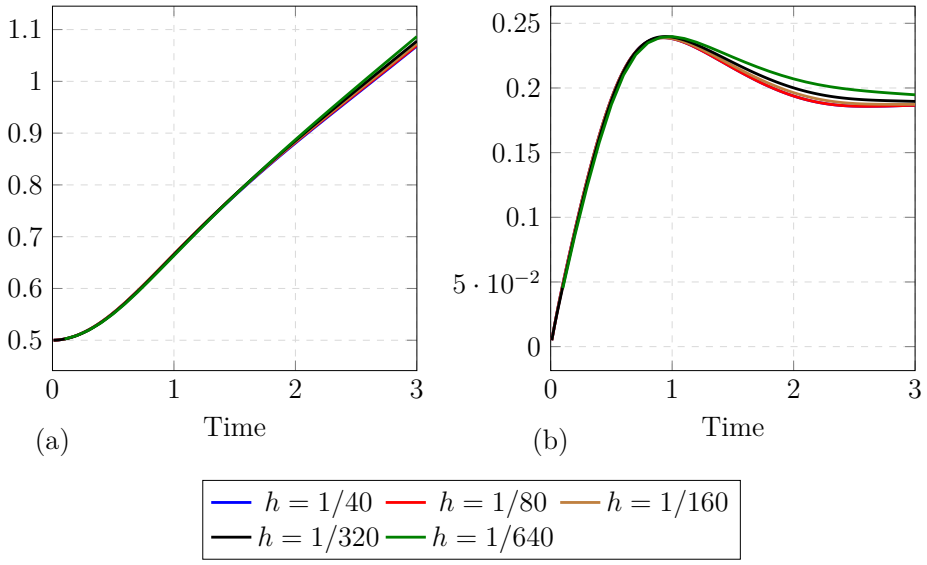


Figure 4.17: Centroid position over time (a), mean rise velocity (b) calculated for VoF method at different mesh resolutions.

Following on table 4.4 the most characteristic benchmark parameters are presented for each mesh refinement, namely the minimum circularity, the maximum rise velocity and the time at which these are tracked and finally the final centroid position. As it is possible to derive also from table 4.5 and from charts 4.16 and 4.17 the deviations are limited and the simulations converge towards the finest grid as the ROC computation suggests. The convergence rate is less than linear but it increases with mesh resolution and the relative errors are in the order of $10^{-3}$ for bubble circularity and center of mass and $10^{-2}$ for rise velocity.

Table 4.4: Minimum circularity and maximum rise velocity with corresponding time and centroid final position

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.8933 | 0.8922 | 0.8946 | 0.8984 | 0.9088 |
| $t\|_{c=c_{min}}$ | 1.95 | 1.9297 | 1.9297 | 1.9203 | 1.9 |
| $v_{max}$ | 0.2389 | 0.2389 | 0.2389 | 0.2398 | 0.2398 |
| $t\|_{v=v_{max}}$ | 0.9296 | 0.9203 | 0.9297 | 0.9406 | 0.97 |
| $y_c\|_{t=3}$ | 1.0683 | 1.0697 | 1.0727 | 1.0778 | 1.0865 |

Table 4.5: Relative error norms and convergence orders for VoF.

| $h$ | $\|\|e\|\|_1$ | $ROC_1$ | $\|\|e\|\|_2$ | $ROC_2$ | $\|\|e\|\|_\infty$ | $ROC_\infty$ |
|---|---|---|---|---|---|---|
| *Circularity* | | | | | | |
| 1/40 | 0.00852 | | 0.164 | | 0.00153 | |
| 1/160 | 0.0068 | 0.2596 | 0.127 | 0.2716 | 0.00112 | 0.2989 |
| 1/320 | 0.00447 | 0.6053 | 0.0836 | 0.6004 | 0.00744 | 0.5928 |
| *Center of Mass* | | | | | | |
| 1/40 | 0.00716 | | 0.139 | | 0.0168 | |
| 1/160 | 0.00539 | 0.2926 | 0.1 | 0.3039 | 0.0127 | 0.2838 |
| 1/320 | 0.00336 | 0.6795 | 0.0619 | 0.6969 | 0.00801 | 0.6655 |
| *Rise Velocity* | | | | | | |
| 1/40 | 0.0411 | | 0.164 | | 0.0636 | |
| 1/160 | 0.0328 | 0.2596 | 0.127 | 0.2716 | 0.0465 | 0.2990 |
| 1/320 | 0.0215 | 0.6054 | 0.0836 | 0.6004 | 0.0309 | 0.5927 |

## 4.2 `twoPhaseEulerFoam`

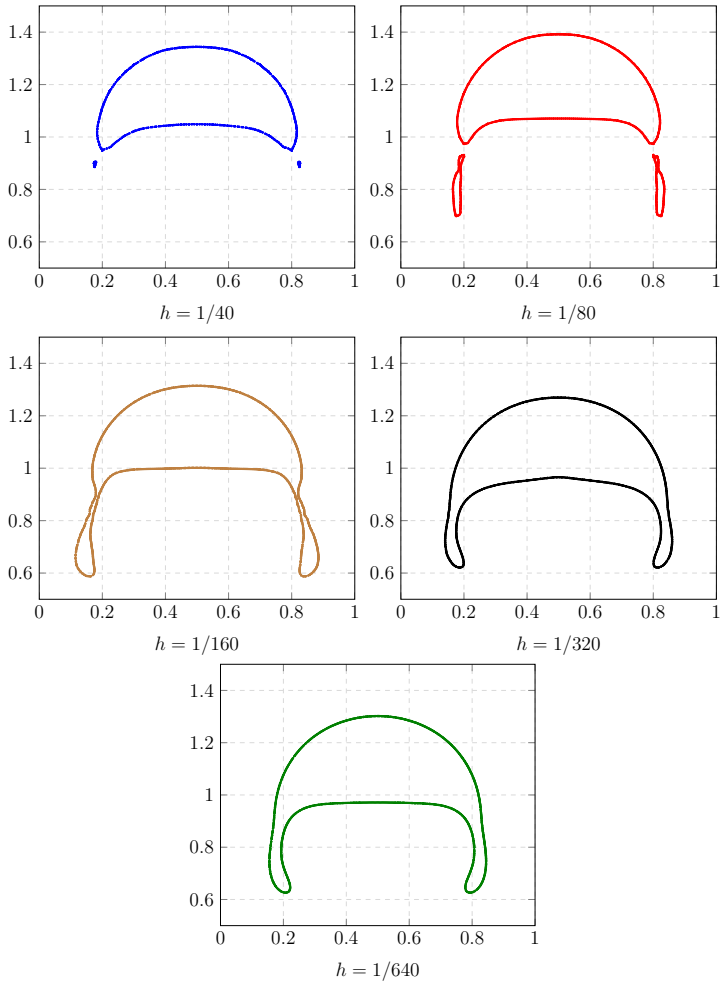


Figure 4.18: Visual confrontation of contours at last time-step for different mesh refinements.

In this section the same analysis performed on the VoF solver `interFoam` was carried on the two-fluid solver `twoPhaseEulerFoam`. As mentioned in the introductory part of this section these calculations were aimed to detect whether or not the two-fluid method, as implemented in Open$\nabla$Foam, is able to capture phase interface and if it is, how correctly compared to the VoF methodology (that was already proven as a good benchmark reference).
Following the bubble contours at the last time-step are reported for all the consecutive mesh refinements to provide the reader with a visual confronta-

tion.

Figure 4.19 represents bubble circularity over time. As in the previous case

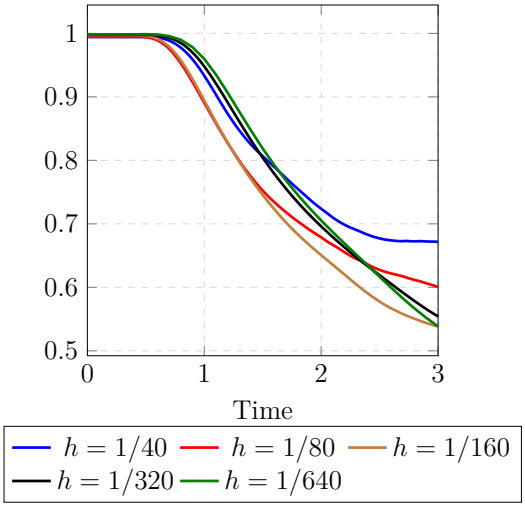


Figure 4.19: bubble circularity over time calculated for two-fluid method at different mesh resolutions.

study the first 0.5 time units present a constant value of bubble circularity, which decreases monotonically towards the minimum reached at the final time-step, this absolute minimum ranges between a centred neighbourhood of 0.6. The excursion in bubble circularity is quite big and this is evident in figure 4.18, a breakup is spotted in the two coarsest mesh while in the finest simulations the characteristic "tails" of the skirted bubble regime are tracked. This denotes a highly unphysical solutions for all the mesh refinements proposed in this study, `twoPhaseEulerFoam` tracks a skirted interface in a case, as defined in the introduction, that is characterized by dimensionless numbers typical of ellipsoidal bubble range. This conclusion is supported also by 4.20 where respectively centroid position over time (a) and the mean bubble rise velocity (b) are reported. A diverging trend is captured for bubble centroid position, while bubble mean rise velocity shows a similar trend but with different final values and maxima positions. The only similarity can be spotted in the two most refined solution which present the same trend.

Following on table 4.6 the most characteristic benchmark parameters are presented for each mesh refinement, namely the minimum circularity, the maximum rise velocity and the time at which these are tracked and finally the final centroid position. As it is possible to derive also from table 4.7 and from charts 4.19 and 4.20 the deviations are important and the simulations

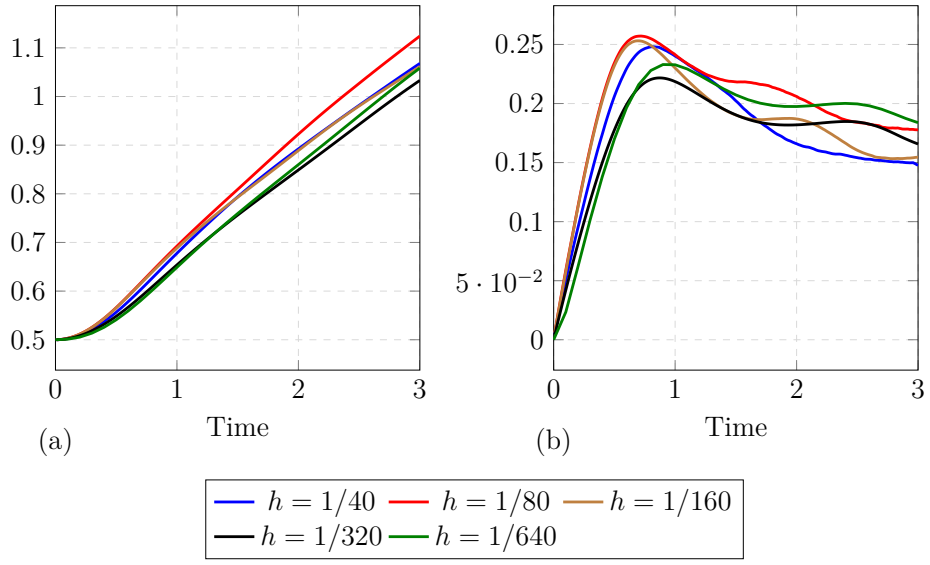


Figure 4.20: Centroid position over time (a), mean rise velocity (b) calculated for VoF method at different mesh resolutions.

hardly converge towards the finest grid as the ROC computation suggests. These are symptoms of an intrinsic error committed by the solver, the more the mesh is refined the more the simulation tends to provide a cohesive bubble with a highly different bubble circularity. That's why no convergence is reached towards the finest grid solution (diverging more than linear ROC) and no correspondence to VoF or literature [13] can be detected, as will be highlighted in the following section.

Table 4.6: Minimum circularity and maximum rise velocity with corresponding time and centroid final position

| $h$ | 1/40 | 1/80 | 1/160 | 1/320 | 1/640 |
|---|---|---|---|---|---|
| $c_{min}$ | 0.6713 | 0.6006 | 0.5383 | 0.5542 | 0.5385 |
| $t\|_{c=c_{min}}$ | 3 | 3 | 3 | 3 | 3 |
| $v_{max}$ | 0.2481 | 0.2571 | 0.2531 | 0.2217 | 0.233 |
| $t\|_{v=v_{max}}$ | 0.8203 | 0.71 | 0.6906 | 0.8703 | 0.9 |
| $y_c\|_{t=3}$ | 1.0681 | 1.1243 | 1.0616 | 1.0331 | 1.0585 |

Table 4.7: Relative error norms and convergence orders for VoF.

| $h$ | $\|\|e\|\|_1$ | $ROC_1$ | $\|\|e\|\|_2$ | $ROC_2$ | $\|\|e\|\|_\infty$ | $ROC_\infty$ |
|---|---|---|---|---|---|---|
| *Circularity* | | | | | | |
| 1/40 | 0.0413 | | 0.0597 | | 0.1331 | |
| 1/160 | 0.0496 | 0.1625 | 0.0587 | 0.1266 | 0.0788 | -0.0059 |
| 1/320 | 0.0107 | -2.2115 | 0.0123 | -2.256 | 0.0157 | -2.326 |
| *Center of Mass* | | | | | | |
| 1/40 | 0.0303 | | 0.0321 | | 0.0329 | |
| 1/160 | 0.0325 | -0.9438 | 0.0349 | -0.9093 | 0.0372 | -0.7590 |
| 1/320 | 0.0128 | -1.3486 | 0.0154 | -1.1794 | 0.0240 | -0.6341 |
| *Rise Velocity* | | | | | | |
| 1/40 | 0.1415 | | 0.1564 | | 0.1954 | |
| 1/160 | 0.1494 | 0.4258 | 0.17 | 0.21 | 0.2756 | -0.0405 |
| 1/320 | 0.0774 | -0.9488 | 0.078 | -1.1235 | 0.0834 | -1.7248 |

## 4.3 Comparison

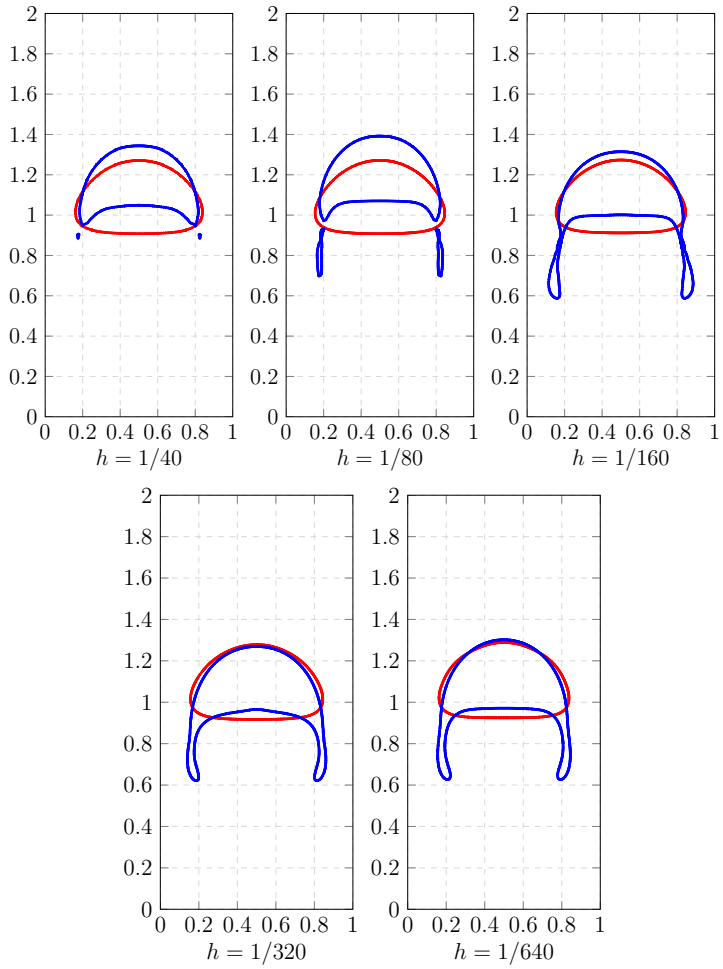


Figure 4.21: Visual comparison between two-fluid method (blue) and VoF (red) of contours at $t = 3$ for different mesh resolutions.

Figure 4.21 reports how the interface is transported, at the last time-step, using the same mesh resolution for the two solvers. It is evident, in view of the above, that the two solutions are not comparable. As highlighted in `interFoam` results' comment section the solution (red contours) is in line with both theory (2.1) and other benchmark simulations [13]. On the other hand `twoPhaseEulerFoam` results (blue contours) give a clear view on how this solver is not capable to track, in this bubble shape regime, the interface as correctly as the VoF solver. The coarsest simulations' lack of resolution

can be detected as the cause of the bad interface capturing.

But with mesh refinement no improvement is visible, on the contrary for finer meshes ($h = 1/160$, $h = 1/320$ and $h = 1/640$) a skirted shape is detected.

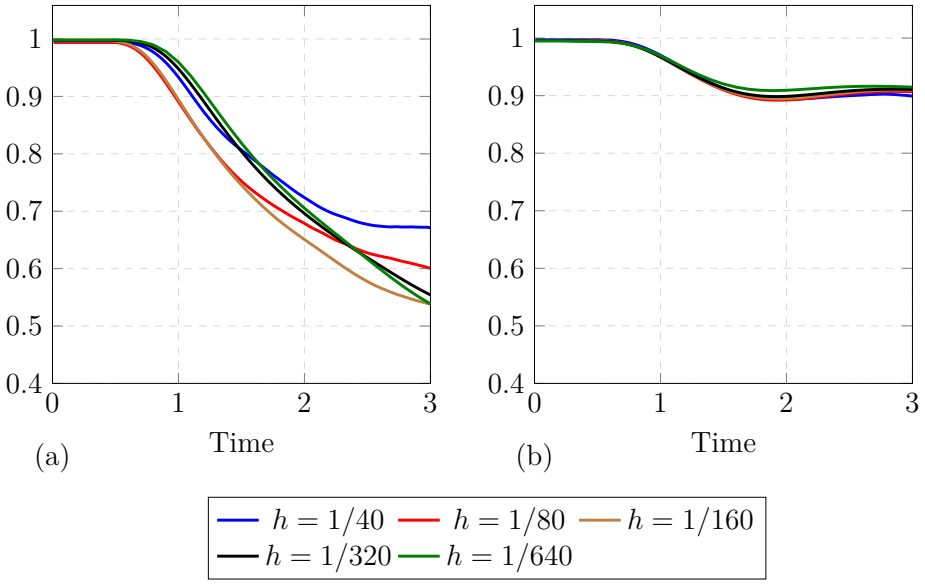


Figure 4.22: comparison between bubble circularities at different mesh resolutions for both two-fluid method (a) and VoF (b).

As it is possible to see from figures 4.22,4.23 and 4.24 the other benchmark quantities are not totally comparable. Starting from bubble circularity it can be said that the evolution trend is similar in the two methodologies, namely for the first 0.5 time units the circularity remains almost constant and then decreases monotonically and finally, in the last time-steps, the slope decreases to reach a plateau. As said before in this section the two solutions are not comparable at any mesh refinement level. This is evident from figure 4.22 that an unphysical interface tracking is captured by the two-fluid model. The final values of circularity are not comparable at all and that is reflected by the contours reported before (Fig. 4.21). This results are consistent with what said earlier for the bubble contours and support the theory of a problem of interface capturing in *twoPhaseEulerFoam.*

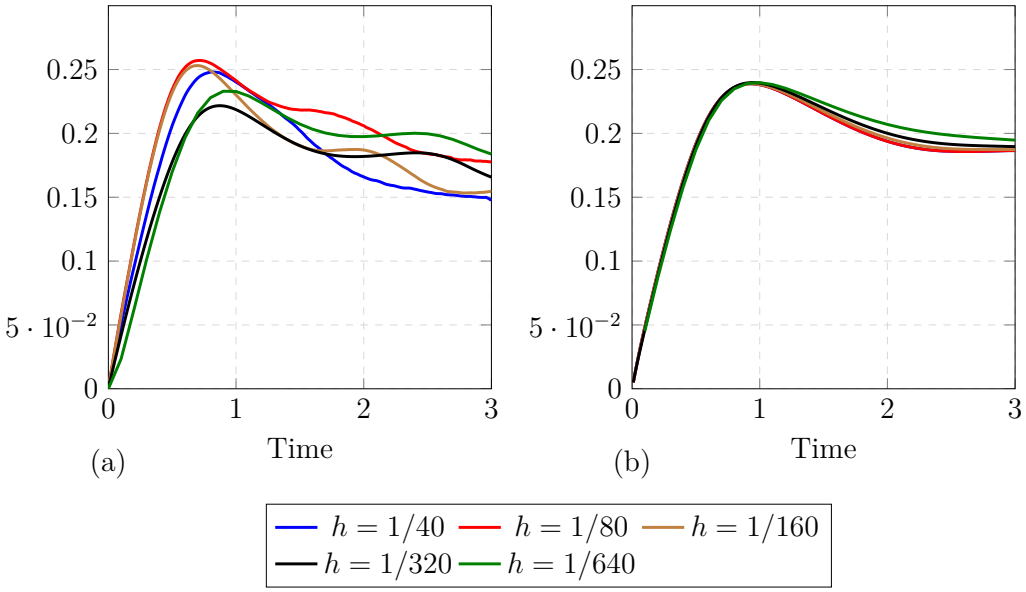


Figure 4.23: comparison between bubble rise velocities at different mesh resolutions for both two-fluid method (a) and VoF (b).

For what concerns the other two benchmark quantities (mean bubble rise velocity and bubble centroid position) we can spot a behaviour similar to the one highlighted in 3.3. Namely, with mesh refinement both velocity and centroid position trends tend to align to the VoF solution. The bubble velocity is tracked more or less correctly with mesh refinement, the curve shape is similar to VoF's one, as said for Case 2 with the formation of bubble tails the mean bubble rise velocity is affected with a localized concavity change (in time). The peak position is also affected by mesh resolution but with refinement the position and maximum value tend to the VoF solution. In fact, as can be seen in figure 4.21 the bubbles with mesh refinement tend to superpose. The deviations reported in figure 4.23 are consistent with the contours described before, even though the velocity field is correct in the most refined meshes the bubble centroid deviates with respect to VoF solutions because of the unphysical breakup tails tracked by two-fluid model. This confirm what said before for "Case 2" that bubble momentum equation is correctly solved, meaning that the velocity field is correct, while the interface is evidently incorrect.

The analysis on case 1 allowed to confirm what was only possible to assume from case 2 results. Namely, the two-fluid methodology as implemented in OpenFOAM through `twoPhaseEulerFoam` takes correctly into account the two separate momentum equations for the two segregated phases but it is not

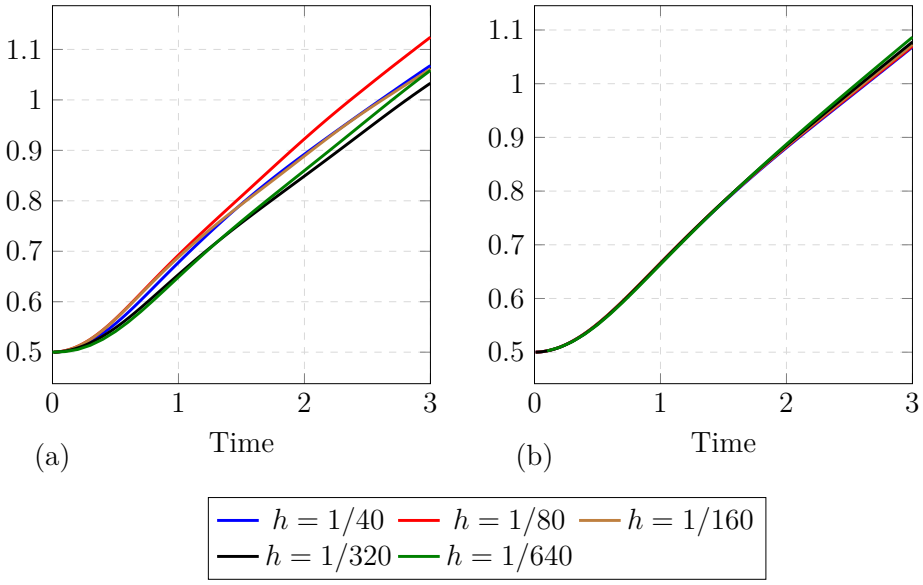


Figure 4.24: comparison between bubble centroids at different mesh resolutions for both two-fluid method (a) and VoF (b).

capable to track in a physical way the interface between phases. This lack of resolution in interface tracking is evident from both case-study highlighting that neither bubble breakup nor ellipsoidal-interface can be modelled. Especially in case 1, that is a surface tension dominated regime, it is clear that something is missing in the solver. The solution lacks of cohesion, the bubble as it evolves through time tend to collapse and high diffusion of the interface is captured. The reasons of this behaviour will be analysed in the following chapter as part of the conclusion of This work.

# Chapter 5

# Closure

A feasibility study for the application of a two-fluid CFD methodology for simulating two-phase cavitating flows. In this study, two-fluid and VoF interface-capturing methodologies have been discussed and compared. A thorough description of state of the art numerical techniques as well as mathematical models of key physical phenomena has been carried on. This last Chapter concludes the thesis and is divided in two Sections. The first summarises the conclusions acquired from this work, especially from the results derived in the previous chapter, and the second one makes suggestions for future work.

## 1 Conclusions

Recalling what said in the introductory chapter about the requirements of a solver to be suitable for simulating cavitation in injectors and injector-like geometries, a two-fluid solver should capture gas/liquid interface correctly and be able to take into account the slip velocity between phases. Since cavitation is the most important source/sink phenomenon in this kind of problems, the solver should be capable to model also this contribution. Starting from the latter requirement this section will describe the procedure followed during this thesis research work and will highlight the principal conclusions.

### 1.1 Cavitation

As described in Chapter 3 `reactingTwoPhaseEulerFoam` features a "*interfaceCompositionModel*" that allow to determine through a user-selected model the composition of the species at phase-interface. Cavitation is a phenomenon of phase change driven by pressure, i.e. if a local depression happens a portion

of liquid could evaporate if if local pressure falls behind saturation pressure at fluid's temperature. Since the transport of mass fractions is supported and the bound between species transport source/sink term and volume fraction transport equation source/sink term was demonstrated in chapter 3, the to introduction of a homogeneous reactor model for cavitation in a two-fluid solver can be perform by taking advantage of the high modularity of the existing framework.

## 1.2 Interface Tracking

The core-subject of this work was to study interface capturing abilities of the two-fluid solver as implemented in OpenFOAM-dev. In chapter 4 a thorough analysis based on two benchmark case studies was carried on in order to quantify both qualitatively and quantitatively the interface-tracking quality of the method. The results brought to the author's attention that not only the two fluid Euler-Euler methodology is more mesh demanding than the VoF solver, used as benchmark solver since already validated as a reliable one in literature, but also that severe diffusion and lack of cohesion manifest even at high mesh resolutions. This suggest that a modelling error is committed systematically by the solver. Recalling what said about segregated-flows modelling in chapter 2, it appears evident that two main contributions define the bubble shape and consequently the bubble evolution[1] over time:

- Drag Force;
- Surface Tension Force

The derivation of the drag force term was discussed in chapter 2. On the other hand, solutions of Case 1 simulations show a lack of cohesion of the bubble, which shows a skirted profile with a set-up typical of ellipsoidal range bubbles, while simulations of Case 2 show a lack of cohesion in coarsest meshes tat manifest in an early breakup followed by a unphysical interface tracking. This suggests that in *twoPhaseEulerFoam* the form of the terms for surface tension is not suitable for the case studied.

An accurate revision of the solver's code resulted in the finding that no equation for surface tension is currently implemented in OpenFOAM-dev. This somehow justifies the irrational results obtained in chapter 4, where a physically more accurate solver provided in some case unphysical solutions and in other cases a low-quality solution. As repeatedly pointed out surely

[1] Both bubble shape and trajectory/velocity

the solution acquires physical meaning with mesh refinement, but that is the demonstration that surface tension forces are not being taken into account. in fact, recalling chapter 2, it was demonstrated that Euler-Euler segregated solvers capture the interface of liquid-gas phases thanks to the combined action of two factors:

- Mesh Resolution;
- Surface Tension source term for momentum equation

From the benchmark cases it is evident that with mesh refinement a more physical-like bubble is described, meaning that the behaviour of the solver is in line with the theoretical one. This can be seen as a further demonstration that the surface tension source term is not modelled in OpenFOAM-dev, and hence that the two-fluid solver, as it is, is not suitable for segregated flows simulations.

## 1.3 Slip Velocity

Although some diffusion in the interface tracking was found, the momentum equations an hence the velocity fields resulted generally consistent with the benchmark case and the solution of the VoF solver. The mean rising velocity and bubble centroid position for each case study tend to converge towards the finest solutions, which are comparable in terms of velocity magnitude and peaks position. This demonstrates that with the two fluid method it is possible to obtain comparable solutions (from the velocity point of view) but with a double accurate information, i.e. instead of having a "mixture velocity" and a "slip velocity", where the latter is modelled, we will have two transport equation solving for gas and liquid velocities and, as a consequence, the slip velocity will be calculated from the "exact" fluids velocities. This feature is the reason why the Euler-Euler methodology was investigated as possible solution for this kind of problems. Thinking about the phenomena that this feasibility study aims to allow studying, such as vortex cavitation in injectors [5], it is straightforward to see how the two fluid method is less dependent on modelling and, hence would provide a more physical solution of the velocity field.

# 2 Suggestions for Future Work

Starting from the conclusions derived in the previous section, the principal suggestions for future work development are described, two main aspects will be considered: cavitation and interface tracking.

## 2.1 Cavitation

Several options can be implemented; the simplest barotropic model for cavitation, coupled to a homogeneous reactor model, should be implemented and the solver should be validated through a pilot study [7] and also compared to the Vof solver (`interPhaseChangeFoam`) solution. Once the cavitation phenomenon is demonstrated to be correctly tracked by the solver, a further development of the cavitation modelling could be studied. In particular more complex models can be implemented, namely models such as Delannoy, Koop and Schmidt [54–56] that use a transition function providing a smoother passage from a phase to the other with pressure change, as showed in figure 5.1. Together with this improvement in barotropic cavitation mod-

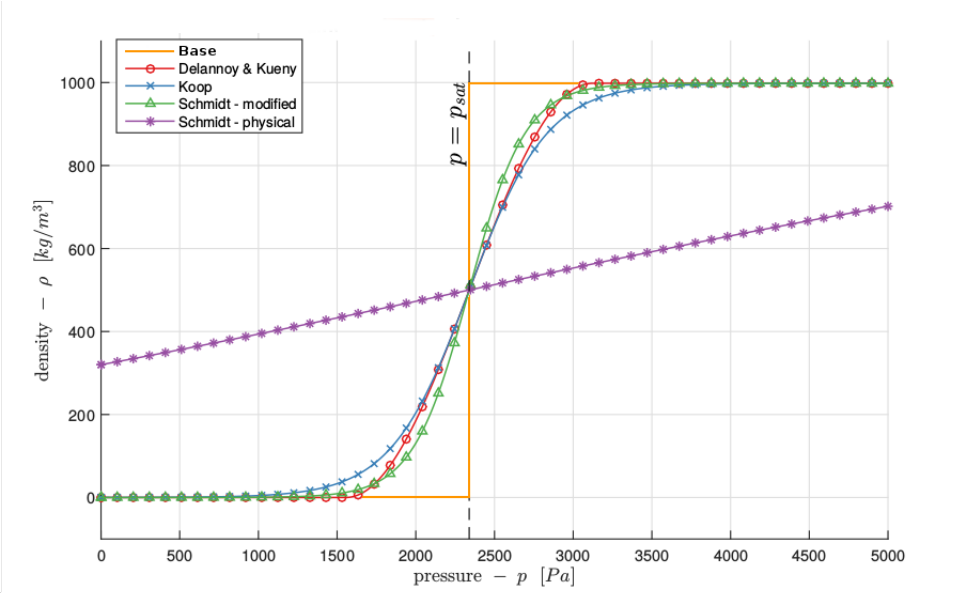


Figure 5.1: Different barotropic models for water at T=293 K

elling other, more complex, models should be considered. In literature, as well as in OpenFOAM, models based on the Rayleigh-Plesset equation such as Schnerr and Sauer were demonstrated to be reliable to describe cavitation and widely validated

[18].

## 2.2 Interface Tracking

Interface tracking in two fluid solver should be improved. It is suggested as a further development to implement the surface tension source term described in 2.68. The implementation should be consistent with the OpenFOAM formulation of two fluid method, in particular the source term should be activated only if the segregated version of the solver is being used. It could be added a section in *phaseProperties* dictionary where it is possible to activate the source term, in this way `twoPhaseEulerFoam` could be used both for dispersed flows and segregated ones. For what concerns the validation of the upgraded solver it is recommended to follow the benchmark case [13] as well as the analysis described in chapter 4 of this thesis.

# Appendix A

# The & Operator and *divU* derivation

## 1 Discretisation of PDEs

During discretization, the partial differential equations are integrated over each element in the mesh resulting in a set of algebraic equations with each one linking the value of the variable at an element to the values at its neighbours. The algebraic equations are then assembled into global matrices and vectors and the coefficients of every equation stored at the row and column locations corresponding to the various element indices (Fig A.2). The linear system that derives from this discretisation has the form of:

$$\mathbf{A}\vec{x} = \vec{b} \tag{A.1}$$

Where $\mathbf{A}$ represents the matrix of coefficients, $\vec{x}$ is the field that will be derived solving the system and finally $\vec{b}$ represents the effect of both source terms and boundary conditions.
Taking a deeper look at the matix of coefficients every cell has one and only one diagonal coefficient, none of which are zero. Therefore the diagonal list contains N values, where N is the number of cells in the mesh.The upper and lower triangles have (N - 1)N possible values, however most are zero. The non-zero values correspond to cell-pairs that influence one another. Open$\nabla$FOAM uses a small computational molecule, therefore only adjacent cells influence one another. In other words, there are two non-zero coefficients for every internal face in the mesh: one that appears in the lower triangle; and one that appears in the upper triangle. These coefficients are usually opposite (especially when fluxes are involved) and , thus for each face

a neighbouring and an owner cell are defined such that:

$$\mathbf{A}_{i,j} = \begin{cases} Owner & i > j \\ Diagonal & i = j \\ Neighbour & i < j \end{cases}$$

This can be easily seen starting from a simple 3×3 matrix as defined in figure (Fig.A.1):

| | | |
|---|---|---|
| 1 | 2 | 3 |
| 6 | 5 | 4 |
| 7 | 8 | 9 |

Figure A.1: Example matrix.

Applying the procedure described before it is then possible to derive the non zero coefficients in Open∇FOAM:

| i \ J | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 |
|---|---|---|---|---|---|---|---|---|---|
| 1 | X | N | | | | N | | | |
| 2 | O | X | N | | N | | | | |
| 3 | | O | X | N | | | | | |
| 4 | | | O | X | N | | | | N |
| 5 | | O | | O | X | N | | N | |
| 6 | O | | | | O | X | N | | |
| 7 | | | | | | O | X | N | |
| 8 | | | | | O | | O | X | N |
| 9 | | | | O | | | | O | X |

Figure A.2: "X" refears to diagonal elements, "O" refears to owner cells and "N" refears to neighbouring cells

# 2 the calculation of divU

"&" operator is defined in ***fvMatrix.C*** and is used in ***pEqn.C*** to derive **phase$_\varphi$.divU**. In this section the "&" operator will be described and then the formulation of divU will be derived as a function of **pEqnComp$_\varphi$**.

## 2.1 the "&" operator

As it is possible to see from the caller of the function, this operator creates a *GeometricField* starting from an *fvMatrix*, namely **M** and a *DimensionedField*, in this case **psi**.

```
template<class Type>
Foam::tmp<Foam::GeometricField<Type, Foam::fvPatchField,
    Foam::volMesh>>
Foam::operator&
(
const fvMatrix<Type>& M,
const DimensionedField<Type, volMesh>& psi
)
```

Dimensions of the GeometricField are set as dimensions of matrix **M** divided by volume regardless of which manipulations will be done afterwards on the matrix.
Remembering what said about the discretization of PDEs through a system of linear equations (Equation (A.1)) and considering that in Open$\nabla$FOAM that exact system of discretised equations is represented through an fvMatrix, it is possible to define various sub-matrices starting from the linearized system of equation **M**.
It is possible to define briefly **M** as:

$$\mathbf{M} := \mathbf{A_M}\vec{x} = \vec{b} \tag{A.2}$$

This identity allows the definition of the diverse matrix composing **M**:

**M.diag**: represents the diagonal part of matrix $\mathbf{A_M}$;

**M.source**: represents the explicit vector of system **M**, namely $\vec{b}$;

**M.H**:represents the H operator in *lduMatrix*.

### 2.1.1 H operator in lduMatrix

Before continuing the description of "&" operator it is necessary to focus on the definition in Open$\nabla$FOAM of the **H** operator.
Adressing of elements in the matrix is fulfilled through ***lduMatrix***, two poiters are defined to navigate through te elements of the matrices:

**uPtr:** namely the upper adress, for upper list it represents column number while for lower list represents row number;

**lPtr:** namely the lower adress, for upper list it represents row number while for lower list represents column number.

If, for instance, we take into account matrix **A** as defined in equation (A.3):

$$A := \begin{bmatrix} 1 & 0 & 5 & 0 \\ 0 & 2 & 0 & 6 \\ 7 & 0 & 3 & 0 \\ 0 & 8 & 0 & 4 \end{bmatrix} \tag{A.3}$$

It is possible to rewrite it using lduMatrix notation, namely describing it through diagonal, upper and lower elements:

$$\begin{bmatrix} d(0) & - & u(0) & - \\ - & d(1) & - & u(1) \\ l(0) & - & d(2) & - \\ - & l(1) & - & d(3) \end{bmatrix} \tag{A.4}$$

Once this matrix has been defined, the two pointers (**uPtr** and **lPtr**) can be listed as:

**uPtr** = [2,3] Representing the columns of upper elements;

**lPtr** = [0,1] Representing the columns of lower elements.

To have a better understanding of the procedure, it is interesting to procede with the example introduced with equation (A.3). The vector resulting from **H** ooperator is defined as:

```
for (label face=0; face<nFaces; face++)
{
HpsiPtr[uPtr[face]] -= lowerPtr[face]*psiPtr[lPtr[face]];
HpsiPtr[lPtr[face]] -= upperPtr[face]*psiPtr[uPtr[face]];
}
```

Thanks to the definition of uPtr and lPtr it is possible to write:

$$HpsiPtr[2] -= l(0) \cdot \psi(0) \quad HpsiPtr[3] -= l(1) \cdot \psi(1)$$
$$HpsiPtr[0] -= u(0) \cdot \psi(2) \quad HpsiPtr[1] -= u(1) \cdot \psi(3)$$

It is possible to reorder the resulting vector as:

$$HpsiPtr := \{-u(0) \cdot \psi(2); -u(1) \cdot \psi(3); -l(0) \cdot \psi(0); -l(1) \cdot \psi(1)\}^T \tag{A.5}$$

The description of the adressing of lduMatrix allows to interpret the formulation of **H** as defined in ***lduMatrixTemplates.C***:

$$\begin{aligned} H_N- &= A_O \cdot \psi_O \\ H_O- &= A_N \cdot \psi_N \end{aligned} \tag{A.6}$$

Where "N" and "O" refear respectively to neighbouring and owner cells and so to upper and lower triangular matrix.

The Transformation of the **M** matrix throug **H** operator can be graphically summarised as:

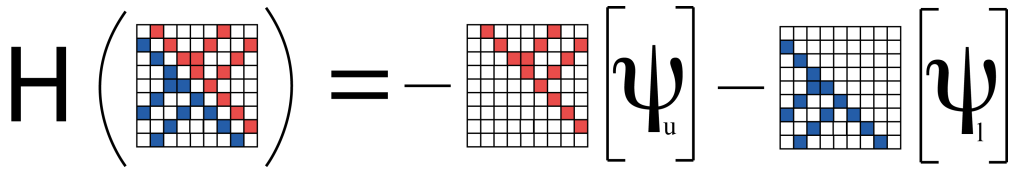


Figure A.3: The H operator

It can be easily demonstrated that the **H** operator represents the opposite inner product of the extra diagonal matrix and a vector, namely:

$$H(\mathbf{A}) := -\mathbf{A_{ED}} \cdot \vec{\psi} \tag{A.7}$$

Where $\mathbf{A_{ED}}$ is the extra diagonal matrix.

It is now possible to write the formulation of the "&" operator as defined in Open$\nabla$FOAM:

$$\mathbf{M}_\varphi = -\frac{-[\mathbf{M.diag} + BC_i \cdot \mathbf{I}] \cdot \psi + \mathbf{M.H}(\psi) + \mathbf{M.source} + BC_e}{V} \tag{A.8}$$

Where $BC_i$ represents the diagonal matrix of implicit boundary conditions and $BC_e$ represents the vector of explicit boundary conditions.
As it is possible to infer from equation (A.8) the dimensions given to the GeometricField by the "&" operator are consistent with its formulation.

## 2.2 The derivation of "divU"

**pEqnComp** represents the compressible part of the pressure equation, taking into account both compressibility of the phases and mass transfer. This

factor is made by summing an explicit contribution to an implicit underelaxed one. The reasons behind this definition are due to the bounding of the solution.
This compressible part of the pressure equation is then added to the incompressible part and is solved to derive the pressure field $\mathbf{P_{rgh}}$. Once obtained the solution of the pressure equation is possible to derive explicitly the term **divU** through the "&" operator, as seen in Chapter 3. This procedure is summarised graphically in figure A.4, from which is possible to highlight also the dimensions consistency.

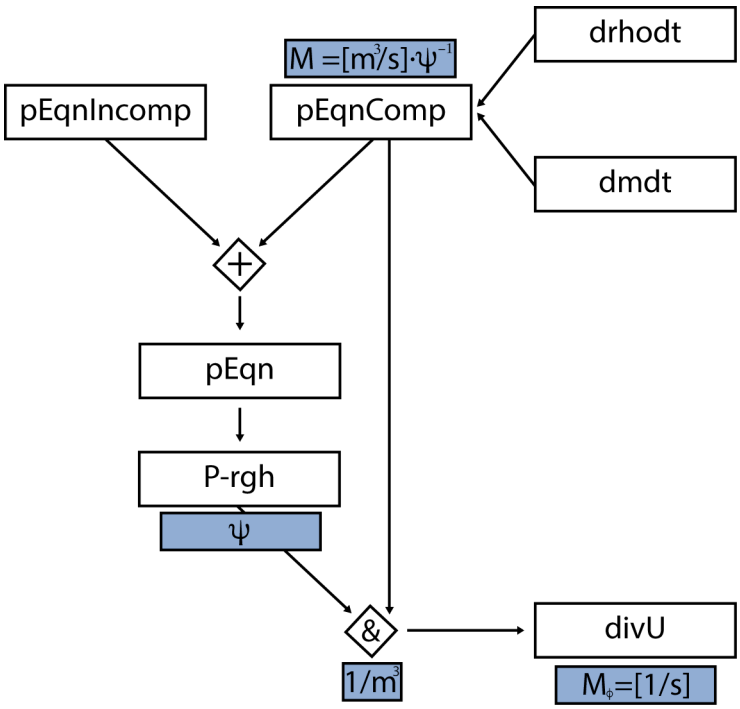


Figure A.4: Schematic representation of *divU* derivation

# Appendix B

# Interface-Composition Models

## 1 Lift Coefficient Models

Different models are available in literature for the calculation of lift coefficients, the most commonly used will be described in the following sections.

### 1.1 Moraga Model

It is applicable mainly to the lift force on spherical solid particles, though it can be applied to liquid drops and bubbles. In this model the lift coefficient combines opposing actions of two phenomena:

- Classical aerodynamic lift resulting from the interaction between the dispersed phase particles and the primary phase shear;
- The vorticity-induced lift resulting from interaction between particles and vortices shed by particle wakes.

As a result, the lift coefficient is defined in terms of both particle Reynolds number and vorticity Reynolds number:

$$Re_p = \frac{\rho_q |\vec{v}_q - \vec{v}_p| d_p}{\mu_q} \tag{B.1}$$

$$Re_\omega = \frac{\rho_p |\nabla \times \vec{v}_q| d_p^2}{\mu_q} \tag{B.2}$$

Introducing $\phi = Re_p Re_\omega$, The Moraga lift coefficient is defined as [57]:

$$C_l = \begin{cases} 0.0767 & \phi \leq 6000 \\ 0.2 e^{\frac{\phi}{3.6} \times 10^{-5} - 0.12} \cdot e^{\frac{\phi}{3} \times 10^{-7}} & 6000 < \phi < 5 \times 10^7 \\ -0.6353 & \phi \geq 5 \times 10^7 \end{cases} \tag{B.3}$$

### 1.2 Legendre-Magnaudet Model

Is applicable primarily to small diameter spherical fluid particles, though it can be applied to non-distorted liquid drops and bubbles. The Legendre-Magnaudet model accounts for the momentum transfer between the flow around the particle and the inner recirculation flow inside the fluid particle caused by the fluid friction/stresses at the fluid interface. Therefore, the predicted lift force coefficients are smaller than for rigid solid particles. The range of validity given by Legendre and Magnaudet is [58]:

$$0.1 < Re_p < 500 \tag{B.4}$$

$$St = 2\beta \leq 1 \tag{B.5}$$

$$\beta = 0.5(re_\omega / Re_p) \tag{B.6}$$

And with $St = \frac{fL}{\vec{v}}$ the Strouhal number. The lift coefficient can be written as:

$$C_l = \sqrt{(C_{l,lovRe})^2 + (C_{l,highRe})^2} \tag{B.7}$$

With:

$$C_{l,lowRe} = \frac{6}{\pi^2}(Re_p St)^{-0.5}\frac{2.55}{(1+0.2\frac{Re_p}{St})^{1.5}} \tag{B.8}$$

$$C_{l,highRe} = \frac{1}{2}\frac{1+16Re_p^{-1}}{1+29Re_p^{-1}} \tag{B.9}$$

## 2 Wall Lubrication Models

Different models are available in literature for the calculation of lubrication coefficient, the most commonly used will be described in the following sections.

### 2.1 Antal Model:

Antal was the first to use this force to predict the near wall peak void fraction. This model computes the lubrication coefficient as [59]:

$$C_{wl} = max\left(0, \frac{C_{w1}}{d_b} + \frac{C_{w2}}{y_w}\right) \tag{B.10}$$

Where $C_{w1} = -0.01$ and $C_{w2} = 0.05$ are non dimensional coefficients, $d_b$ is the bubble diameter and $y_w$ is the distance to the nearest wall. It is

interesting to note that the lubrication coefficient is different from zero only within a thin layer adjacent to the wall, the region satisfies this relation:

$$y_w \leq -\left(\frac{C_{w2}}{C_{w1}}\right) d_b \tag{B.11}$$

Corresponding to a region $y_w \leq 5d_b$ with default values of reference coefficients. As a result, the Antal model will only be active on a sufficiently fine mesh and grid independence can only be achieved with very fine meshes.

## 2.2 Tomiyama Model

The expression for the lubrication coefficient for this model is [46, 60]:

$$C_{wl} = C_w \frac{d_b}{2}\left(\frac{1}{y_w^2} - \frac{1}{(D-y_w)^2}\right) \tag{B.12}$$

Where $D$ represents the pipe diameter and $C_w$ depends on the Eötvos number ($Eo$).

$$C_w = \begin{cases} 0.47 & Eo < 1 \\ e^{-0.933Eo+0.179} & 1 \leq Eo \leq 5 \\ 0.00599Eo - 0.0187 & 5 < Eo \leq 33 \\ 0.179 & 33 \leq Eo \end{cases} \tag{B.13}$$

Where $Eo = \frac{g(\rho_q - \rho_p)d_b^2}{\sigma}$ and $\sigma$ is the surface tension.
This model is superior to Antal's [59] but it is restricted to flows in pipe due to the dependence on pipe diameter.

## 2.3 Frank Model

The model developed by Frank et al. [61] removes the dependence on pipe diameter in the Tomiyama model, expressing the lubrication coefficient as:

$$C_{wl} = C_w max\left(0, \frac{1}{C_{wd}} \cdot \frac{1 - \frac{y_w}{C_{wc}d_b}}{y_w\left(\frac{y_w}{C_{wc}d_b}\right)^{m-1}}\right) \tag{B.14}$$

where:

$C_w$: is a function of $Eo$ as in Equation (B.13);

$C_{wd}$: is the damping coefficient and determines the relative magnitude of the force;

$C_{wc}$: is the cut-off coefficient and determines the distance to the wall within which the force is active;

$m$: is the power law constant ($1.5 \leq m \leq 2$).

# 3 Drag Models

Different models are available in literature for the calculation of Drag coefficient, the most commonly used will be described in the following sections.

## 3.1 Tomiyama Model

This model is well suited to gas-liquid flows in which the bubbles can have a range of shapes. The drag function is modeled as in the Shiller and Neumann Modell (Equation (2.48)). But in this case the formulation of the drag coefficient ($C_D$) is different [62, 63]:

$$C_D = max\left(min\left(\frac{24}{Re}(1 + 0.15Re^{0.687}), \frac{72}{Re}\right), \frac{8}{3}\frac{Eo}{Eo + 4}\right) \tag{B.15}$$

The Eotvos number ($Eo$) is defined in section 2.2.

## 3.2 Ishii-Zuber Model

This model, as implemented in Open$\nabla$FOAM, distinguishes two regimes with conseguent different drag modeling:

- The spherical bubble regime with a drag coefficient defined by the Ishii-Zuber formula for spherical solid particles [64]:

$$C_d(sphere) = max\left(\frac{24}{Re_{bm}}(1 + 0.15Re_{bm}^{0.687}), 0.44\right) \tag{B.16}$$

where the Reynolds number is defined as:

$$Re_{bm} = \frac{\rho_{bubble}|\vec{U}_1 - \vec{U}_2|D_s}{\mu_m} \tag{B.17}$$

$$\mu_m = \mu_{bubble}\left(1 - \frac{\alpha_a}{\alpha_{max}}\right)^{-2.5\alpha_{max}\mu^*} \tag{B.18}$$

$$\mu^* = \frac{\mu_1 + 0.4\mu_2}{\mu_1 + \mu_2} \tag{B.19}$$

This model is valid only for solid particles in the undistorted regime [Rushe]. $\alpha_{max}$ is the maximum phase fraction and depends on the maximum packing value considered. It is user-defined and Rusche [15] suggested a value of 0.62 for solid particles and 1.0 for fluid particles (if the Ishii-Zuber model valid for bubbles is used).

- The ellipse distorted regime:

$$C_d(ellipse) = E(\alpha_1)C_{d\infty} \tag{B.20}$$

where:

$$C_{d\infty} = \frac{2}{3}Eo^{1/2} \tag{B.21}$$

$$E(\alpha_1) = \frac{1 + 17.67f(\alpha_1)^{(6/7)}}{18.67f(\alpha_1)} \tag{B.22}$$

$$f(\alpha_1) = \frac{\mu_2}{\mu_m}(1 - \alpha_1)^(1/2) \tag{B.23}$$

And the Eötvös number is defined as the ratio between gravitational and surface tension forces:

$$Eo = \frac{(\rho_2 - \rho_1)gD_S^2}{\sigma} \tag{B.24}$$

An automatic regime selection is performed in the code according to [2]:

$$C_d = C_d(sphere) \quad \text{if} \quad C_d(sphere) \geq C_d(ellipse) \tag{B.25}$$

$$C_d = C_d(ellipse) \quad \text{if} \quad C_d(sphere) < C_d(ellipse) \tag{B.26}$$

# 4 Turbulent Dispersion Models

The following sections describe the models for turbulent dispersion force.

## 4.1 Lopez de Bertodano Model

Instead of following equation (2.51), Lopez de Bertodano proposed this formulation, one of the simplest available in literature [65, 66]:

$$\vec{F}_{td,q} = -\vec{F}_{td,p} = C_{TD}\rho_q k_q \nabla\alpha_p \tag{B.27}$$

Where:

$\rho_q$: continuous phase density:

$k_q$: turbulent kinetic energy of continuous phase:

$\nabla\alpha_p$: gradient of dispersed phase volume fraction:

$C_{TD}$: modifiable variable (initially set as 1).

## 4.2 Burns Model

This model is based on the Favre averaging of the drag term [66]:

$$\vec{F}_{td,c} = -\frac{3}{4}C_d\frac{\rho_d}{D_s}\frac{\nu_d^t}{\sigma_t}|\vec{U}_r|\alpha_c\left(\frac{1}{\alpha_c}+\frac{1}{\alpha_d}\right)\nabla\alpha_c \tag{B.28}$$

where $\sigma_t = 0.9$and $C_d$ stands for the drag force coefficient that is used during the simulations.

## 4.3 Gosman Model

This is the formulation of the Gosman et al. [39] model as implemented in Open$\nabla$FOAM:

$$\vec{F}_{td,c} = -\frac{3}{4}C_d\frac{\rho_d}{D_s}\frac{\nu_d^t}{\sigma_t}|\vec{U}_r|\nabla\alpha_c \tag{B.29}$$

where $\sigma_t = 0.9$ and $C_d$ stands for the drag mpdel coefficient.

# Bibliography


[1] NUCLEAR ENERGY AGENCY. Proceedings of the cfd for nuclear reactor safety applications (cfd4nrs-3) workshop. In *Proceedings of the CFD for Nuclear Reactor Safety Applications (CFD4NRS-3) Workshop*, 2010.

[2] Alberto Ghione. Development and validation of a two-phase cfd model using openfoam, 2012.

[3] Ahmadreza Ghaffarkhaha, Mohammadjavad Ameri Shahrabia, and Mostafa Keshavarz Moravejib. Application of cfd for designing conventional three phase oilfield separator. *Egyptian Journal of Petroleum*, 2017.

[4] Daniel Sebastia-Saez, Sai Gu, Panneerselvam Ranganathan, and Konstantinos Papadikis. Micro-scale cfd modeling of reactive mass transfer in falling liquid films within structured packing materials. *International Journal of Greenhouse Gas Control*, 2015.

[5] Federico Piscaglia and Jerome Hélie. On the evolution of confined string cavitating vortices in nozzle holes and their influence on spray atomisation. In *CTR Summer Program, Center for Turbulence Research, Stanford University, USA*, 2016.

[6] Matthieu Dreyer. *Mind The Gap: Tip Leakage Vortex Dynamics and Cavitation in Axial Turbines*. PhD thesis, ÉCOLE POLYTECHNIQUE FÉDÉRALE DE LAUSANNE, 2015.

[7] Baris Bicer and Akira Sou. Application of the improved cavitation model to turbulent cavitating flow in fuel injector nozzle. *Applied Mathematical Modelling*, 2015.

[8] E. Winklhofer, E. Kull, E. Kelz, and A. Morozov. Comprehensive hydraulic and flow field documentation in model throttle experiments under cavitation conditions. In *ILASS-Europe 2001*, 2001.

[9] Akira Sou, Baris Bicer, and Akio Tomiyama. Numerical simulation of incipient cavitation flow in a nozzle of fuel injector. *Computers and Fluids*, 2014.

[10] W. Edelbauer. Numerical simulation of cavitating injector flow and liquid spray break-up by combination of eulerian–eulerian and volume-of-fluid methods. *Computers and Fluids*, 2016.

[11] Mohammad Passandideh-Fard and Ehsan Roohi. Transient simulations of cavitating flows using a modified volume-of-fluid (vof) technique. *International Journal of Computational Fluid Dynamics*, 2008.

[12] Alexander Morozov and Uwe Iben. Experimental analysis and simulation of cavitating throttle flow. In *6th International Conference on Heat Transfer, Fluid Mechanics and Thermodynamics*, 2008.

[13] S. Hysing, S. Turek, D. Kuzmin, N. Parolini, E. Burman, S. Ganesan, and L. Tobiska. Quantitative benchmark computations of two-dimensional bubble dynamics. *International Journal for Numerical Methods in Fluids*, 2008.

[14] H. Weller. Derivation, modelling and solution of the conditionally averaged two-phase flow equations. Technical report, OpenCFD, 2005.

[15] H. Rusche. Computational fluid dynamics of dispersed two-phase flows at high phase frac- tions. phd thesis, 2002.

[16] H. K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics*. Pearson, 1995.

[17] J.H. Ferziger and M. Peric. *Computational Methods for Fluid Dynamics*. Springer, 2002.

[18] Kaushik Saha and Xianguo Li. Assessment of different cavitation models in mixture and eulerian framework for two-phase flow in diesel injectors. In *Proceedings of the ASME 2013 Internal Combustion Engine Division Fall Technical Conference*, 2013.

[19] Holger Marschall. *Towards the Numerical Simulation of Multi-Scale Two-Phase Flows*. PhD thesis, Technische Universität München, 2011.

[20] ANSYS. *ANSYS Fluent Theory Guide Release 17.0*. © SAS IP.

[21] MAmoru Ishii and Takashi Hibiki. *Thermo-Fluid Dynamics of Two-Phase Flow*. Springer, 2011.

[22] M. Abdulkadir. *Experimental and computational fluid dynamics (CFD) studies of gas-liquid flow in bends.* PhD thesis, University of Nottingham, 2011.

[23] C. E. Brennen. Fundamentals of multiphase flows. *Cambridge University Press*, 2005.

[24] V.Hernandez-Perez. *Gas-liquid two-phase flow in inclined pipes.* PhD thesis, University of Nottingham, 2007.

[25] B.J. Azzopardi. Gas-liquid flows. *Begell House*, 2006.

[26] Y. Taitel, D. Barnea, and A.E. Dukler. Modelling flow pattern transitions for steady upward gas-liquid flow in vertical tubes. *AICHE. J.*, 1980.

[27] E.S. Rosa, B.F. Flora, and M.A.S.F. Souza. Design and performance prediction of an impedance void meter applied to the petroleum industry. *Measurement Science and Technology*, 2012.

[28] R. Clift, J. R. Grace, and M. E. Weber. Bubbles, drops and particles. *Academic Press, London, 1st edition*, 1978.

[29] S. Elgobashi. On predicting particle-laden turbulent flows. *Appl. Sci. Res., 52*, 1994.

[30] C. Kralj. *Numerical Simulation of Diesel Spray.* PhD thesis, Imperial College, University of London, 1993.

[31] Murray H. Protter and Charles Bradfield Morrey. *Intermediate Calculus.* Springer, 1985.

[32] Hazewinkel and Michiel. *Ostrogradski formula, Encyclopedia of Mathematics.* Springer Science+Business Media B.V. / Kluwer Academic Publishers, 1994.

[33] Williamson, Richard Hale, and Trotter Hale. *Multivariable Mathematics.* Pearson Education, Inc, 2004.

[34] H.Weller. Phase compressible turbulence models.

[35] Sherif H. El Tahry (General Motors Research Laboratories). k-$\epsilon$ equation for compressible reciprocating engine flows. *Journal of Energy*, 7(4), 1983.

[36] Pope S.B. *Turbulent flows*. Cambridge University Press, 2010.

[37] Richard T. Lahey Jr. The simulation of multidimensional multiphase flows. *Nuclear Engineering and Design*, 2005.

[38] E Michta. Modeling of subcooled nucleate boiling with openfoam, 2011.

[39] A.D. Gosman, R. I. Issa, C. Lekakou, M.K. Looney, and S.Politis. Multidimensional modelling of turbulent two-phase flows in stirred vessels. *AIChE J.*

[40] A. Tomiyama, Tamai, H., Zun I., and Hosokawa S. Transverse migration of single bubbles in simple shear flows. *Chemical Engineering Science*, 57(11):1849–1858, June 2002.

[41] Schiller L. and Naumann A. A drag coefficient correlation. *Z. Ver. Deutsch. Ing*, 77:318–320, 1935.

[42] H. Lamb. *Hydrodynamics*. Cambridge University Press, 1895.

[43] Mamoru Ishii, Seungjin Kim, and Joseph Kelly. Development of interfacial area transport equation. *Nuclear Engineering and Technology*, 37(6):525–536, December 2005.

[44] Vakhrushev I.A. and Efremov G.I. Interpolation formula for computing the velocities of single gas bubbles in liquids. *Chemistry and Technology of Fuels and Oils*, 6(5):376–379, May 1970.

[45] Wellek R.M., Agrawal A.K., and Skelland A.H.P. Shape of liquid drops moving in liquid media. *International Journal of Multiphase Flow*, 12(5):854–862, September 1966.

[46] M. Otromke. Implementation and comparison of correlations for interfacial forces in a gas-liquid system within an euler-euler framework, 2003.

[47] *User Guide v 5.0*. Open CFD Ltd., 2017.

[48] Prof. Mohamed El-Amin. *Mass Transfer in Multiphase Systems and its Applications*. InTech, 2011.

[49] Antoine C. Tensions des vapeurs; nouvelle relation entre les tensions et les températures. *Computes Rendus des Séances de l'Académie des Sciences*, 1888.

[50] Buck A.L. New equations for computing vapor pressure and enhancement factor. *Journal Applied Meteorology*, 1981.

[51] C. M. Rhie and W. L. Chow. A numerical study of the turbulent flow past an isolated airfoil with trailing edge separation. *AIAA J*, 1983.

[52] Santiago Marquez Damian. *An extended mixture model for the simultaneous treatment of short and long scale interfaces*. PhD thesis, Universidad Nacional del Litoral, 2013.

[53] Wadell H. Sphericity and roundness of rock particles. *Journal of geology*, 1933.

[54] Y. Delannoy. Two phase flow approach in unsteady cavitation modelling. *ASME Cavitation and Multiphase flow forum*, 1990.

[55] A. H. Koop. *Numerical simulation of unsteady three-dimensional sheet cavitation*. PhD thesis, University of Twente, 2008.

[56] D. P. Schmidt. *Cavitation in diesel fuel injector nozzles*. PhD thesis, University of Wisconsin-Madison, 1997.

[57] Moraga F.J., Bonetto F.J., and Lahey R.T. Lateral forces on spheres in turbulent uniform shear flow. *International Journal of Multiphase Flow*, 25(6-7):1321–1372, September 1999.

[58] Legendre D. and Magnaudet J. The lift force on a spherical bubble in a viscous linear shear flow. *Journal of Fluid Mechanics*, 368:81–126, August 1998.

[59] Antal S.P., Lahey Jr, R.T., and Flaherty J.E. Analysis of phase distribution in fully developed laminar bubbly two-phase flow. *International Journal of Multiphase Flow*, 17(5):635–652, September 1991.

[60] Tomiyama A. Struggle with computational bubble dynamics. *Multiphase Science and Technology*, 10(4):369–405, 1998.

[61] Frank T. Advances in computational fluid dynamics (cfd) of 3-dimensional gas-liquid multiphase flows. *NAFEMS Seminar: Simulation of Complex Flows (CFD)*, pages 1–18, April 2005.

[62] Tomiyama A., Kataoka I., and Sakaguchi T. Drag coefficients of bubbles (1st report, drag coefficients of a single bubble in a stagnant liquid). 1995.

[63] Tomiyama A., Kataoka I., and Sakaguchi T. Drag coefficients of bubbles (2nd report, drag coefficients for a sarm of bubbles and its applicability to transient flow). 1995.

[64] Mamoru Ishii and Novak Zuber. Drag coefficient and relative velocity in bubbly, droplet or particulate flows. *AIChE Journal*, 25(5):843–855, September 1979.

[65] Lopez de Bertodano M. Turbulent bubbly two-phase flow in a triangular duct.

[66] Burns A.D., Frank T., Hamill I., and Shi J.M. The favre averaged drag model for turbulent dispersion in eulerian multi-phase flows. *5th international conference on multiphase flow*, 4(392), May 2004.