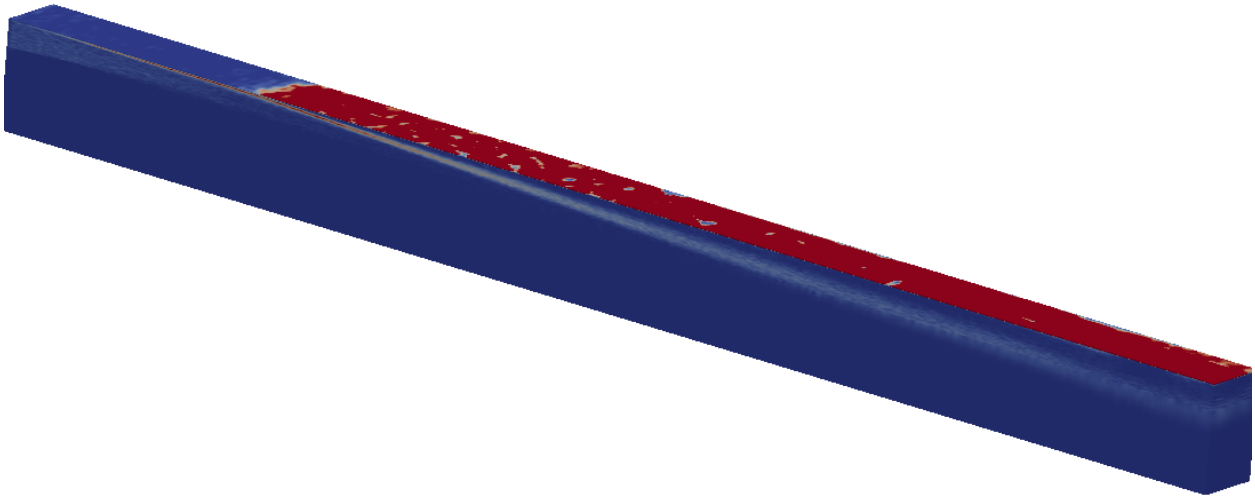
CHALMERS
UNIVERSITY OF TECHNOLOGY



Numerical investigation of bubbles in channel flow

Implementation and analysis of the influence of bubbles injection on the drag of a ship

Master's thesis in Nordic Master in Maritime Engineering

RÉMY LE GUEN

Department of Shipping and Marine Technology
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2016

Master's thesis X-16/353

Numerical investigation of bubbles in channel flow

Implementation and analysis of the influence of bubbles injection on the drag of a ship

RÉMY LE GUEN

CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Shipping and Marine Technology
Chalmers University of Technology
Gothenburg, Sweden 2016

Numerical investigation of bubbles in channel flow
Implementation and analysis of the influence of bubbles injection on the drag of a ship
RÉMY LE GUEN

Supervisor:
RICKARD BENSOW, Department of Shipping and Marine Technology
SVERRE STEEN, Institutt for marin teknikk, NTNU

Master's Thesis 2016:X-16/353
Department of Shipping and Marine Technology
Chalmers University of Technology
SE-412 96 Gothenburg
Telephone +46 31 772 1000

Cover: Result of a 3D-simulation performed in a Lagrangian framework for bubbles of 1mm diameters.

Typeset in LaTeX
Printed by Reproservice
Gothenburg, Sweden 2016

Numerical investigation of bubbles in channel flow
Implementation and analysis of the influence of bubbles injection on the drag of a ship

RÉMY LE GUEN
Department of Shipping and Marine Technology
Chalmers University of Technology

Abstract

This master's thesis aims to study the numerical methods that can simulate the influence of injection of air bubbles under a ship's hull. The geometry of the hull is simplified to a flat plate and the analysis is only done in two dimensions. An emphasis is given into the prediction of the reduction of the viscous resistance that is observed in experiments. This study is carried out using CFD analysis with the software OpenFOAM.
Two frameworks, Eulerian-Eulerian and Eulerian-Lagrangian, that allows the simulation of a multiphase flow are set-up and the assumptions behind all the required models are explained. The simulations predicts a gain of efficiency between 0% and 15% depending on the diameter of the bubbles. Some differences in the results produced by the two different methods are highlighted and guidance is given on which solver to use for a given case depending on the diameter and the concentration of the bubbles below the plate.

Keywords: multiphase simulation, OpenFOAM, drag reduction, Lagrangian Particle Tracking.

Acknowledgements

First I would like to thanks my supervisor Prof. Rickard Bensow to have helped me setting-up the subject of the thesis and guiding me through the process. I want also to thank PhD student Sankar Menon Cherubala Pathayapura to have helped me for the practical matters all along the work.

To Prof. Sverre Steen for having been my NTNU supervisor and for having given me the interest in naval hydrodynamics during my first year of master.

I would also like to thank the staff of the "École Centrale de Lyon" (and especially the international office staff) who gave me the opportunity to realize this double degree.

To Prof. Poul Andersen for coordinating this amazing programme that is the Nordic Master in Maritime Engineering and for helping during the problem encountered.

Enfin, j'aimerais remercier mes amis et ma famille pour m'avoir supporté dans cette aventure.

Le Guen Rémy, Gothenburg, June 2016

Contents

List of Figures

List of Tables

Nomenclature

Latin symbols

Symbol	Dimension	Description
C_d	–	Drag coefficient
C_F	–	Friction skin coefficient
C_l	–	Lift coefficient
C_{td}	–	Turbulent diffusion coefficient
C_{vm}	–	Virtual mass coefficient
C_{wl}	–	Wall lubrication coefficient
D	[m]	Diameter of the bubble
D_s	[m]	Sauter diameter of the bubble
$\vec{g}$	$[\mathrm{m.s^{-2}}]$	Gravity vector
k	$[\mathrm{m^2.s^{-2}}]$	Turbulent kinetic energy
$\vec{n}$	–	Wall normal vector
u	$[\mathrm{m.s^{-1}}]$	Turbulent fluctuation of the velocity
$\vec{U}$	$[\mathrm{m.s^{-1}}]$	Velocity
$\overline{U}$	$[\mathrm{m.s^{-1}}]$	Mean velocity
$\vec{U}_r$	$[\mathrm{m.s^{-1}}]$	Relative velocity $\vec{U}_r = \vec{U}_b - \vec{U}_f$
y	[m]	Near wall distance
y^+	–	Dimensionless distance from the wall

Greek symbols

Symbol	Dimension	Description
α	–	Void fraction
ε	$[\mathrm{m^2.s^{-3}}]$	Turbulent energy dissipation
μ	[Pa.s]	Dynamic viscosity
ν	$[\mathrm{m^2.s^{-1}}]$	Kinematic viscosity
ν^T	$[\mathrm{m^2.s^{-1}}]$	Turbulent kinematic viscosity
ρ	$[\mathrm{kg.m^{-3}}]$	Density
τ	$[\mathrm{N.m^{-2}}]$	Shear stress
σ	–	surface tension
Φ	$[\mathrm{m^{-3}}]$	Source/sink in the IAC equation
ϕ	–	Aspect ratio of a bubble

Adimensional numbers

Symbol	Dimension	Description
Re_b	–	Reynold Bubble number
Eo	–	Eotvos number
We	–	Weber number
Mo	–	Morton number

Subscripts

Symbol	Description
$_b$	Bubble phase
$_f$	Fluid phase
$_w$	Wall

Acronyms

Acronym	Description
CFD	Computational Fluid Dynamics
IAC	Interface area concentration
LPT	Lagrangian Particle Tracking
MPPIC	MultiPhase Particle-In-Cell method
MULES	Multidimensional Universal Limiter with Explicit Solution
RANS	Reynolds-averaged Navier–Stokes equations

1

Introduction

Due to environmental concerns and the rising fuel cost, the shipping industry aims for a better fuel efficiency for their vessels. One of the way to achieve this is to reduce the resistance of the ships. The frictional resistance – the dominant resistance component for low-Froude number ships – is difficult to reduce as it strongly depends on the wetted surface of the ship. Thus, the interest in techniques that reduce the frictional resistance has increased over the last two decades and several research projects in the USA, Europe and Asia have investigated the possibility of reducing frictional drag by using air lubrication. This technique can be divided into three major categories (Ceccio and Simo, 2011) as illustrated in figure 1.1: Bubble Drag Reduction (a); Air Layer Drag Reduction (b) and Partial Cavity Drag Reduction (c). This thesis will only be focused on dispersed bubbles in the flow.

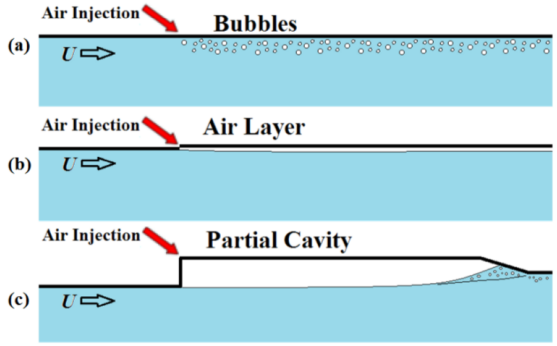


Figure 1.1: The three major categories for air lubrication (extracted from Ceccio and Simo, 2011, fig. 0.1)

In 2011, the Mitsubishi Air Lubrication System (MALS) was the first bubble drag reduction system in the world to be applied to a newly built ship, and is said by the company to have resulted in a substantial reduction in the ship's resistance. Likewise, a project about air cavity ships is currently carried out by Chalmers [1] and aims to "study the optimum configuration of a stable air cavity with the least drag force and air flow rate through experimental investigation in water tunnel and computational fluid dynamics (CFD) technique".

In figure 1.1.c, the closure of the air cavity involves the creation and the ejection of air bubbles. In a way, the presence of bubbles can be beneficial and lead to an

[1] `https://www.chalmers.se/en/projects/Pages/Energy-Efficient-Air-Cavity-Ships.aspx`

additional reduction of the drag but it can also lead to a loss of efficiency if the bubbles reduce the thrust of the propeller.

There are uncertainties about the typical gain of efficiency expected from the injection of bubbles along the hull. If some experiments predict a reduction of 50 % – and sometimes even up to 80 % – in the viscous resistance (Sanders et al., 2006; Kawabuchi et al., 2011), others researches only find a marginal value (Maritech, 2011). Hence, this technology is still at an early stage of development and numerous experiments and simulations must be carried on.

Experimental measures are not easy to realize: it is difficult to efficiently control the ejection of bubbles (with a constant flow rate and diameter) and the use of a model ship can be troublesome as no recognized scaling method has been developed. Moreover, if the final gain of efficiency can easily be deduced, the position and behaviour of small bubbles are difficult to follow in a basin without disturbing the flow field. Hence, it is difficult to observe the mechanisms behind the behaviour of the bubbles just by experiments and the simulations are useful to help for this. The simulation of multiphase flow in the frame of a ship analysis is still a new field and, if there have been numerous simulations and experiments carried out for bubble flow in a pipe and in a vertical water column, very few work has been made in a ship-configuration (ie. in open water and with a predominant effect of the gravity). However, some experiments exist for horizontal pipes (Yoshida et al., 1998; Pang et al., 2014) and can be used as a basis for the comparison of the results.

The prime difficulty of these types of simulation are the large difference of scales: the flow's characteristic length can be of the order of meters but are directly influenced by details of the particle–bubbles interactions, which take place on a millimetre scale. To describe the hydrodynamics of both the gas and particle phase, two main type of models have been developed: the Eulerian-Eulerian and Eulerian-Lagrangian models. The turbulence of the flow will be modelled with the Reynolds-averaged Navier–Stokes (RANS) equations. The simulation will be realized with the open-source software OpenFOAM which has numerous advanced solver for multiphase simulation.

In the first part of the thesis, a framework for the simulation of a multiphase fluid below a ship is developed. One of the goals is to give a guideline to easily set-up a multiphase simulation in the frame of a resistance analysis. In a second part, the numerical implementation of the models in OpenFOAM is discussed and a brief description of the methods used by the solvers is presented to counteract the lack of documentation provided by the software. Finally, the simulation of a bubbly-flow in the boundary layer of a flat-plate is realized in 2-dimensions and for a case as close as possible of the ejection of bubbles after an air cavity. The behaviour of these bubbles and their impact on the resistance will be studied and a special emphasis will be given to the choice of the type of model to use depending on the configuration of the case.

2

Mathematical formulation

This section describes the different models and frameworks that need to be used to set-up a dispersed multiphase flow simulation. First, the possible shapes of the bubbles are discussed (section 2.1) and the equations that characterize the phases are written (section 2.2). The modelling of the interactions between the phases (section 2.3) as well as the turbulence of the flow (section 2.4 are also discussed. Finally, in section 2.5, the models that predict the breakage and coalescence processes are described.

2.1 Shape of the bubble

In a stagnant flow, a bubble is submitted to a gravity/buoyancy force and its cohesion is maintained by the surface tension σ. In this configuration, the most stable shape of a bubble is a sphere. However, submitted to a viscous turbulent hydrodynamic flow, the inertia force cannot be neglected and can modify the form of the interface of the bubble. Hence, the influence of the surrounding liquid to the shape of the bubble must be studied.

2.1.1 The bubble's regimes

The shape of the bubble is closely related to the interaction with the surrounding liquid and the extent of disturbance in the surrounding flow field. Bhaga and Weber (1981) listed and classified several possible shapes for a bubble in a Newtonian liquid. This is displayed in figure 2.1. This classification can be simplified by just distinguishing three regimes:

- The spherical regime: the bubble is considered to be a perfect sphere and thus is fully characterized by its diameter D. It corresponds to the sketch (s) in fig.2.1.
- The ellipsoidal regime: the bubble is considered as an oblate ellipsoid (an ellipsoid with two equal semi-axes) and is characterized by its diameter and aspect ratio ϕ (defined as the ratio of its major axis to its minor axis). It corresponds to the sketches (oe) and (oed) in fig.2.1.
- The cap regime including all the others regimes. These shapes are usually not stable and lead easily to breakage in a turbulent flow.

For the conditions studied in the thesis (a turbulent water flow), it can be considered that the bubbles are never in cap regime.

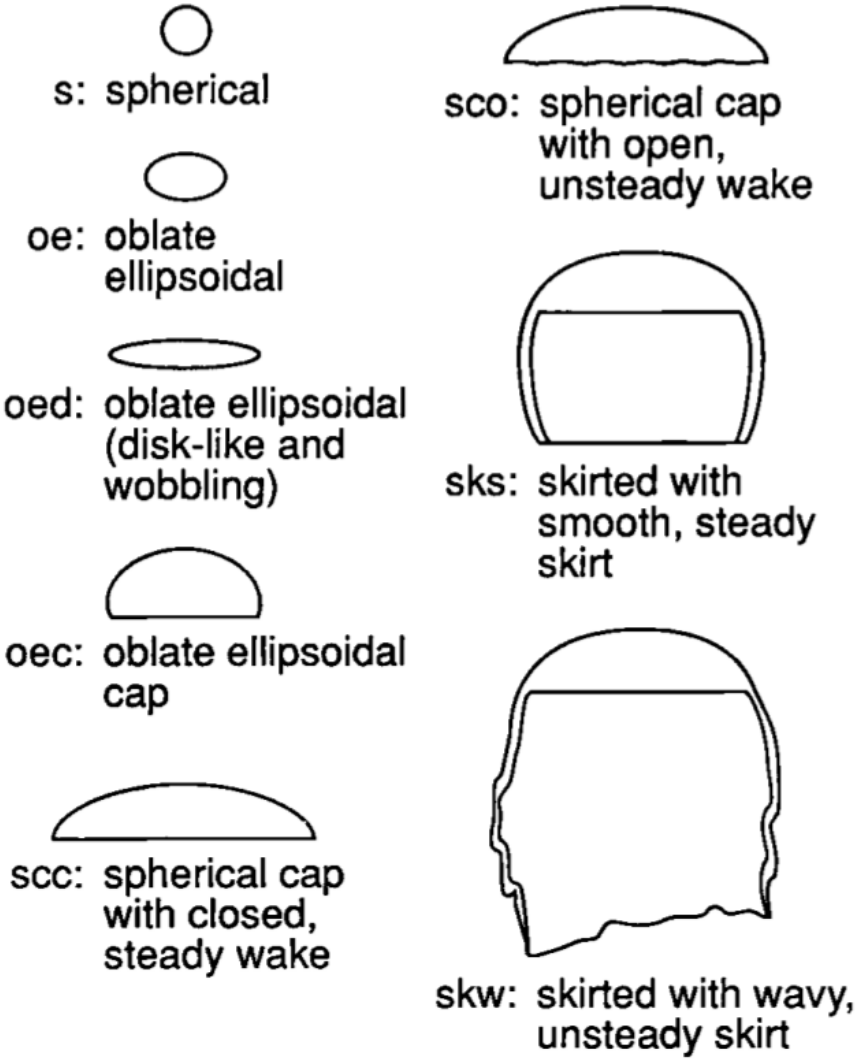


Figure 2.1: Sketches of various bubble shapes observed in infinite Newtonian liquids (extracted from Fan and Tsuchiya, 1990, fig.2.1)

2.1.2 Characterization of the regimes

To characterize the shape of the bubbles moving in a surrounding fluid, some dimensionless numbers are introduced:

- The bubble Reynolds number, Re_b, defined as

$$\mathrm{Re}_b = \frac{U_r D}{\nu_f}, \tag{2.1}$$

  with U_r the relative velocity of the bubble with respect to the fluid.
- The Eötvös number, Eo, (sometimes also called the Bond number Bo) which is the ratio between the gravitational and surface tension forces,

$$\mathrm{Eo} = \frac{(\rho_f - \rho_b) g D^2}{\sigma}, \tag{2.2}$$

  with σ the surface tension of the bubble.

- The Weber number, We, which measures the relative importance of the fluid's inertia compared to its surface tension:

$$\mathrm{We} = \frac{\rho_b U_b^2 D}{\sigma}. \tag{2.3}$$

From these three parameters, another number can be derived: the Morton number Mo:

$$\mathrm{Mo} = \frac{g\mu^4}{\rho\sigma^3} = \frac{\mathrm{We}^2\mathrm{Eo}}{\mathrm{Re}^4}. \tag{2.4}$$

Bhaga and Weber (1981) built a diagram showing the different regimes in function of these three parameters based on experimental observations. This is shown in figure 2.2.

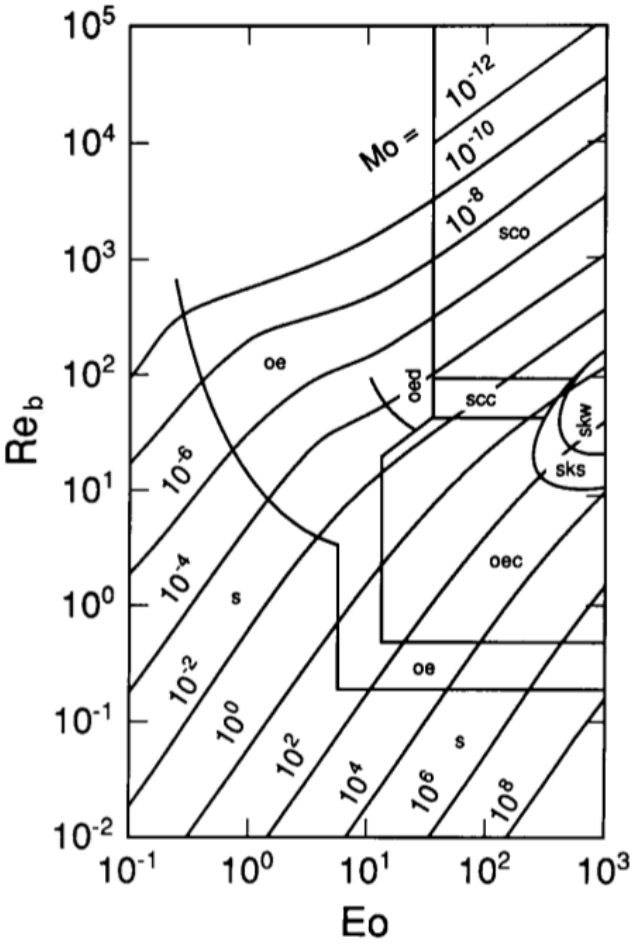


Figure 2.2: Bubble regimes depending on the dimensionless numbers (extracted from Bhaga and Weber, 1981, fig.8)

For fluids with a low Morton number (ie. $\mathrm{Mo} \leq 10^{-3}$), Tadaki and Maeda (1961) found experimentally that the shape of the bubbles in water can be determined from the balance among surface tension, inertial and gravity forces. The Tadaki dimensionless number is introduced:

$$\mathrm{Ta} = \mathrm{Re}_b \mathrm{Mo}^{1/4}. \tag{2.5}$$

From this number:

- the bubble is considered to be spherical if $\mathrm{Ta} \leq 1$;
- the bubble is considered to be an ellipsoid if $1 \leq \mathrm{Ta} \leq 40$;
- the bubble is considered to be in the cap regime if $\mathrm{Ta} \geq 40$.

In ellipsoidal regime, the aspect ratio is also experimentally linked to the Tadaki number. Vakhrushev and Efremov (1970) proposed the relation, valid for $\mathrm{Mo} \leq 10^{-3}$,

$$\phi = \begin{cases} 1 & \text{if } \mathrm{Ta} \leq 1 \\ 0.81 + 0.206 \tanh\left[2\left(0.8 - \log_{10} \mathrm{Ta}\right)\right] & \text{if } 1 \leq \mathrm{Ta} \leq 40 \end{cases} \tag{2.6}$$

In order to easily derive valid relations for spheric and ellipsoid bubbles, an equivalent Sauter diameter D_s is defined as the diameter of a sphere that would have the same volume/surface area ratio as the studied particle. Thus:

$$D_s = \frac{6V_p}{A_p}, \tag{2.7}$$

with V_p and A_p respectively the volume and the surface area of the particle which means that for a sphere $D_s = D$. In all the thesis, when a formula is referring to the bubble's diameter D, it implicitly refers to the Sauter diameter D_s when the bubble is an ellipsoid.

2.2 Dispersed multiphase flow model

The studied flow consists of two phases: the air phase and the water phase. The air phase can be considered as a dispersed phase surrounded by the water phase which is seen as a continuous phase. In a multiphase flow, each of the phases is considered to have a separately defined volume fraction and velocity field but a common pressure field.
In all the thesis, the values referring to the bubbly flow will have the subscript b and the ones referring to the water flow will have the subscript f.
The void fraction for a phase k is defined as

$$\alpha_k = \frac{V_k}{V}, \tag{2.8}$$

where V_k is the volume of the phase k present in the total volume V. Thus, as there are only two phases:

$$\alpha_b + \alpha_f = 1. \tag{2.9}$$

For a dispersed flow, two types of frameworks are prevalent: the Eulerian-Eulerian framework (also called two-fluid model) and the Eulerian-Lagrangian framework (also called Lagrangian Particle Tracking).

2.2.1 Eulerian-Eulerian framework

With the Eulerian-Eulerian framework, the dispersed phase is treated as a second continuous phase interacting with the principal continuous phase. Thus, both phases

are computationally treated as a continuum and are governed by the Navier-Stokes equations.
The Navier-Stokes equations for the bubble phase (characterized by the void fraction α_b and the velocity $\vec{U}_b$) are:

$$\frac{\partial}{\partial t}\left(\alpha_b \rho_b\right) + \nabla \cdot \left(\alpha_b \rho_b \vec{U}_b\right) = 0, \tag{2.10}$$

$$\frac{\partial}{\partial t}\left(\alpha_b \rho_b \vec{U}_b\right) + \alpha_b \rho_b \left(\vec{U}_b \cdot \vec{\nabla}\right) \vec{U}_b = -\alpha_b \vec{\nabla} P + \alpha_b \left(\nabla \cdot \nu_b \vec{\nabla}\right) \vec{U}_b + \vec{M}_b + \rho_b \alpha_b \vec{g}. \tag{2.11}$$

And for the fluid phase (characterized by the void fraction α_f and the velocity $\vec{U}_f$):

$$\frac{\partial}{\partial t}\left(\alpha_f \rho_f\right) + \nabla \cdot \left(\alpha_f \rho_f \vec{U}_f\right) = 0, \tag{2.12}$$

$$\frac{\partial}{\partial t}\left(\alpha_f \rho_f \vec{U}_f\right) + \alpha_f \rho_f \left(\vec{U}_f \cdot \vec{\nabla}\right) \vec{U}_f = -\alpha_f \vec{\nabla} P + \alpha_f \left(\nabla \cdot \nu_f \vec{\nabla}\right) \vec{U}_f + \vec{M}_f + \rho_f \alpha_f \vec{g}. \tag{2.13}$$

$\vec{M}_b$ is the interfacial momentum transfer term and represents the forces that are acting at the interface between the two phases (i.e. the forces acting on a bubble that are caused by the liquid which surrounds it). Therefore according to the Newton's third law of motion:

$$\vec{M}_b + \vec{M}_f = 0. \tag{2.14}$$

The description of the forces acting on the bubble ($\vec{M}_b$) is done in section 2.3.

2.2.2 Eulerian-Lagrangian framework

In an Eulerian-Lagrangian framework, the motion of the dispersed phase is evaluated by following the motion of each bubbles. The water flow is still seen as a continuous phase and thus is still governed by the Navier-Stokes equations:

$$\frac{\partial}{\partial t}\left(\alpha_f \rho_f\right) + \nabla \cdot \left(\alpha_f \rho_f \vec{U}_f\right) = 0, \tag{2.15}$$

$$\frac{\partial}{\partial t}\left(\alpha_f \rho_f \vec{U}_b\right) + \alpha_f \rho_f \left(\vec{U}_f \cdot \vec{\nabla}\right) \vec{U}_f = -\alpha_f \vec{\nabla} P + \alpha_f \left(\nabla \cdot \nu \vec{\nabla}\right) \vec{U}_f + \vec{M}_f + \rho_f \alpha_f \vec{g}. \tag{2.16}$$

The motion of each bubble is driven by the Newton's law:

$$m_b \frac{d\vec{U}_b}{dt} = \vec{M}_b + \rho_b \vec{g}. \tag{2.17}$$

$\vec{M}_b$ represents the forces exerted by the bubbles on the fluid and is described in section 2.3 and m_p is the mass of a bubble which is (in a spherical regime):

$$m_b = \frac{1}{6} \rho_b \pi D^3. \tag{2.18}$$

The concentration of particles influences the interaction between the two phases (Elghobashi, 1991):

- For a very dilute suspension ($\alpha_b \leq 10^{-6}$) the particle's effect on the continuous phase is negligible. Thus, on a first approximation $\vec{M}_f = 0$.
- For a denser suspension ($\alpha_b \leq 10^{-3}$) the particle's effect on the continuous phase is not negligible any more. Usually, $\vec{M}_f$ and must be calculated.
- For dense solution ($\alpha_b \geq 10^{-3}$), the collisions between particles must also be accounted for. These can be done by looking for possible collisions for each particles (as explained in section 3.2.2) or by viewing them statistically with the MPPIC method (as explained in section 2.3.6).

2.2.3 Comparison between the two formulations

The Eulerian-Lagrangian formulation can be considered closer to reality as the bubbles are actually existing in the simulation – unlike the Eulerian-Eulerian method – and therefore would give results closer to reality. This makes also the simulation more intuitive and less dependant on empirical models. However, the computational power needed to track thousands and thousands of particles and to simulate collision between them can be very cumbersome and time consuming.
Thus, the Eulerian-Eulerian method is more suitable for dense solutions as the Lagrangian Particle Tracking would be too computational-intensive. Moreover, for a dense solution, it is quite likely that interactions between particles will be numerous and therefore, a time-averaged model for the interactions can be used quite accurately. However, this method relies a lot on empirical models and hence the quality of the results is strongly dependent on the quality of these models.
It can also be difficult to ensure convergence with the Eulerian-Eulerian method as it is difficult to solve the mass conservation equation (equation 2.10) keeping the boundedness of the void fraction. The Eulerian-Eulerian formulation is really sensitive to the Courant number (Co) and a low number must be ensured (around 0.5). Moreover, a very fine mesh must also be set close to the wall in order to predict the behaviour of the bubbles close to the wall. Therefore the simulation must be performed with very small time-step and thus will increase the computation time.

2.3 The closure term

The interfacial momentum term $\vec{M}_b$ can be broken down into different sub-forces (Nygren, 2014). Some of these forces are usually not included in the Lagrangian formulation.

$$\vec{M}_b = \vec{M}_D + \vec{M}_{VM} + \vec{M}_L + \vec{M}_{WL} + \vec{M}_{TD} + \vec{M}_C, \tag{2.19}$$

with:

- $\vec{M}_D$ the drag force
- $\vec{M}_{VM}$ the virtual mass force
- $\vec{M}_L$ the lift force
- $\vec{M}_{WL}$ the wall lubrication force (usually not included in the Lagrangian formulation)

- $\vec{M}_{TD}$ the turbulent dispersion force when a RANS turbulence model is used (usually not included in the Lagrangian formulation)
- $\vec{M}_C$ the collision force (only included in a MPPIC Lagrangian formulation)

All the individual terms in the interaction force are now described in detail.

2.3.1 Drag force

The drag is the force acting opposite to the bubble motion in the fluid and can be seen as the resistance between the relative motion of the two phases. This force is usually predominant. The drag force is expressed as:

$$\vec{M}_D = -\frac{3}{4}\frac{C_d}{D}\rho_f \alpha_b |\vec{U}_r|\vec{U}_r, \tag{2.20}$$

with C_d the drag coefficient, D the diameter of the bubble and $\vec{U}_r = \vec{U}_b - \vec{U}_f$ the relative velocity between the two phases.
It can be also written with the Reynolds number:

$$\vec{M}_D = -\frac{3}{4}\frac{C_d Re_b}{D^2}\nu_f \rho_f \alpha_b \vec{U}_r = -\frac{3}{4}\frac{C_{dRe}}{D^2}\nu_f \rho_f \alpha_b \vec{U}_r = -K\vec{U}_r \tag{2.21}$$

The expression of C_{dRe} must then be carefully modelled. Two main models have been developed to model the drag experienced by a bubble in a water flow.
The Schiller and Naumann (1935) model is widely used and quite simple but only valid for spherical bubbles. The limit for the spherical regime is set to $\mathrm{Re}_b \leq 1000$ even if it has been seen in section 2.1.2 that it is not a really accurate criterion. For higher Reynolds number, a constant value is set (which does not have any physical meaning but is just set to ensure continuity):

$$C_{dRe} = \begin{cases} 24.0\left(1.0 + 0.15\mathrm{Re}_b^{0.687}\right) & \text{if } \mathrm{Re}_b \leq 1000 \\ 0.44\mathrm{Re}_b & \text{if } \mathrm{Re}_b \geq 1000 \end{cases} \tag{2.22}$$

The Ishii and Zuber (1979) model has extended the Schiller and Naumann model to the elliptic and cap regime:

- In spherical regime, the Schiller and Naumann drag expression is used:

$$C_{dRe}(\text{sphere}) = 24.0\left(1.0 + 0.15\mathrm{Re}_b^{0.687}\right). \tag{2.23}$$

- The ellipse regime is modelled as:

$$C_{dRe}(\text{ellipse}) = \frac{2}{3} f(\alpha_b)\mathrm{Re}_b\sqrt{\mathrm{Eo}}. \tag{2.24}$$

- The cap distorted regime is expressed as:

$$C_{dRe}(\text{cap}) = \frac{8}{3}(1-\alpha_b)^2\mathrm{Re}_b. \tag{2.25}$$

The choice of the regime is determined based on:

$$C_{dRe} = \begin{cases} C_{dRe}(\text{sphere}) & \text{if } C_{dRe}(\text{sphere}) \geq C_{dRe}(\text{ellipse}) \\ \min\left(C_{dRe}(\text{ellipse}), C_{dRe}(\text{cap})\right) & \text{if } C_{dRe}(\text{sphere}) \leq C_{dRe}(\text{ellipse}) \end{cases} \tag{2.26}$$

2.3.2 Virtual mass force

The virtual mass effect occurs when the dispersed phase is accelerated relative to the continuous phase. When this acceleration occurs, part of the surrounding continuous fluid has to be accelerated as well.
The virtual mass force is expressed as:

$$\vec{M}_{VM} = -\rho_f \alpha_b C_{VM} \left(\frac{D_b \vec{U}_b}{Dt} - \frac{D_f \vec{U}_f}{Dt} \right), \tag{2.27}$$

with $\frac{D}{Dt}$ the material derivative defined as:

$$\frac{D_\phi}{Dt} = \frac{\partial}{\partial t} + \vec{U}_\phi \cdot \nabla. \tag{2.28}$$

There are two main models to express the virtual mass coefficient C_{VM}.
One is a constant virtual force model and derives from the application of the potential flow theory to flow around spherical bubbles:

$$C_{VM} = 0.5. \tag{2.29}$$

This model becomes false for non-spherical bubbles so Lamb (1932) proposed an extension for these regimes depending on the aspect ratio ϕ,

$$C_{VM} = \frac{\sqrt{1-\phi} - \phi \arccos \phi}{\phi \arccos \phi - \sqrt{\phi - \phi^2}}. \tag{2.30}$$

2.3.3 Lift Force

The lift force consists of a force acting on bubbles that pushes the bubbles laterally. For a spherical bubble, the lift coefficient C_L is always positive so that the lift force acts towards the wall. However, for deformed larger bubbles, more complicated phenomena arise and an inversion of sign for the lift coefficient is observed in experiments. The force is expressed as:

$$\vec{M}_L = -\alpha_b \rho C_L \overrightarrow{U_r} \wedge \overrightarrow{\nabla} \wedge \vec{U_b}. \tag{2.31}$$

For spherical bubbles, a constant coefficient can be derived from the potential theory:

$$C_L = 0.5. \tag{2.32}$$

If this assumption is not valid, the Tomiyama et al. (2002) model aims to predict the lift force on larger-scale deformable bubbles in the ellipsoidal regime. Its main feature is the prediction of the cross-over point in bubble size at which particle distortion causes a reversal in the sign of the lift force.

$$C_L = \begin{cases} \min\left(0.2888 \tanh\left(0.121 \mathrm{Re}_b\right), f(\mathrm{Eo})\right) & \text{if } \mathrm{Eo} \leq 4 \\ f(\mathrm{Eo}) & \text{if } 4 \leq \mathrm{Eo} \leq 10 \\ -0.27 & \text{if } \mathrm{Eo} \geq 10 \end{cases} \tag{2.33}$$

with $f(\mathrm{Eo}) = 0.0010422\mathrm{Eo}^3 - 0.0159\mathrm{Eo}^2 - 0.0204\mathrm{Eo} + 0.474$.

2.3.4 Wall lubrication force

Experimentally, it was found that the void fraction is often concentrated close to the wall but not touching it (wall-peaked void fraction distribution). This is mainly due to the fact that a bubble close to the wall is likely to rebound at the wall. Hence, the wall lubrication force has been proposed to predict the near wall peak void fraction. The general form of this force is expressed as:

$$\vec{M}_{WL} = \alpha_b \rho_f C_{WL} |\vec{U}_r - (\vec{U}_r.\vec{n})\vec{n}|^2 \vec{n}, \tag{2.34}$$

with $\vec{n}$ the vector normal to the wall.
Frank et al. (2008) proposed a model for C_{WL}:

$$C_{WL} = \begin{cases} f(\text{Eo}) \frac{1-\tilde{y}}{y C_{WD} \tilde{y}^{p-1}} & \text{if } \tilde{y} \leq 1 \\ 0 & \text{if } \tilde{y} \geq 1 \end{cases} \tag{2.35}$$

with $\tilde{y}$ the adimensional value depending on y which is the distance to the wall

$$\tilde{y} = \frac{y}{C_{WC} D} \tag{2.36}$$

and

$$f(\text{Eo}) = \begin{cases} \exp(-0.933\text{Eo} + 0.179) & \text{if } 1 \leq \text{Eo} \leq 5 \\ 0.00599\text{Eo} - 0.0187 & \text{if } 5 \leq \text{Eo} \leq 33 \\ 0.179 & \text{if Eo} \geq 33 \end{cases} \tag{2.37}$$

Thus, C_{WC} is the cut-off coefficient and determines the distance relative to the particle diameter over which the force is active.
The author recommends after extensive testing that $C_{WC} = 10$, $C_{WD} = 6.8$ and $p = 1.7$. Therefore, this involves a very fine mesh close to the wall (as the force is acting just around $10D$).
This force is not included in the Eulerian-Lagrangian formulation as the bubble-wall interaction is taken natively into account in the particle tracking.

2.3.5 Turbulent dispersion force

The turbulent dispersion force accounts for the drag force caused by the turbulent fluctuation of the liquid velocity. Indeed, one can break down the velocity into

$$U = \overline{U} + u \tag{2.38}$$

with $\overline{U}$ the mean velocity and u the turbulent fluctuation. In a RANS simulation, only $\overline{U}$ is simulated while u has also an influence to the closure term. In the Lagrangian formulation, it is usual to model the turbulent velocity based on a stochastic model (this is described in section 2.4.3). However, in an Eulerian-Eulerian model, it is preferable to introduce an additional turbulent dispersion force that models the effect of the turbulent fluctuation of the drag force (which is the predominant force). This is based on the Favre Averaged Drag Model (Burns et al., 2004) and is expressed as:

$$\vec{M}_{TD} = -C_{TD}\vec{\nabla}\alpha_b. \tag{2.39}$$

As this force is proportional to the void-fraction gradient, it can easily generate unstable results.
Lopez de Bertodano (1998) proposed a very simple model to express D:

$$C_{TD} = \rho_f k_f. \tag{2.40}$$

2.3.6 Collision force

The simulation of the collision between particles can lead to long and intensive calculation in a dense particle suspension. Thus, it can be beneficial to account for collision with a statistical approach. It is done by introducing an additional force $\vec{M}_C$ in the interfacial term. This approach is also called the MultiPhase Particle-In-Cell method (MPPIC).
Snider (2001) proposed that:

$$\vec{M}_c = \frac{P_s {\alpha_b}^\beta}{\alpha_{cp} - \alpha_b}. \tag{2.41}$$

The coefficients are chosen based on the work of Patankar and Joseph (2001) who chose $\beta = 3$, $\alpha_{cp} = 0.6$ and $P_s = 100$ Pa. The model only applies to spherical particles and can become not so accurate for areas where the suspension have a low void-fraction.

2.4 Turbulence

For a two-phase flow, the turbulence of the fluid is caused by two main effects: the turbulence that is naturally appearing in the liquid flow at high Reynolds number but also the disturbance created by the bubbles in the fluid.
The turbulence of the bubble phase is assumed to be dependent on the turbulence of the liquid phase through a turbulence response coefficient C_t (defined as the ratio of the root mean square velocity fluctuations of the dispersed phase velocity and of the continuous phase velocity). However, the effect of the turbulence of the bubble-phase can be neglected on a first approach (Rusche, 2002, section 1.5.6). Hence, the bubble phase is considered laminar.
In order to reduce the computation time, wall functions are also used. The region near a wall is not resolved: the first node is located in the log-law region ($30 \leq y^+ \leq 100$) and the flow between the first node is supposed to be as in a single phase boundary layer.

2.4.1 The fluid phase turbulence ($k - \varepsilon$ model)

It is chosen to model the fluid turbulence with a $k - \varepsilon$ model. This model is based on the Reynolds-averaged Navier–Stokes (RANS) equations, meaning only the mean velocity is described and taken into account into the simulation. This allows to neglect the fluctuations of small amplitude and periods and save computational

time. The $k-\varepsilon$ model is also an eddy-viscosity model meaning that the turbulent kinematic viscosity is used to model the effect of the turbulence on the Reynolds stresses in the momentum conservation equation.
The turbulent viscosity is found with the relation:

$$\nu_f^T = C_\mu \frac{k_f^2}{\varepsilon_f} \tag{2.42}$$

with k_f and ε_f respectively the turbulent kinetic energy and the turbulent dissipation of the fluid.
Two differential transport equation are set in order to determine those quantities:

- For turbulent kinetic energy:

$$\frac{\partial k_f}{\partial t} + \frac{\partial (k_f U_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\nu_f + \frac{\nu_f^T}{\sigma_k}\right)\frac{\partial k_f}{\partial x_j}\right] + \nu_f^T S^2 - \varepsilon_f; \tag{2.43}$$

- For turbulent dissipation:

$$\frac{\partial \varepsilon_f}{\partial t} + \frac{\partial (\varepsilon_f U_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\nu_f + \frac{\nu_f^T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon_f}{\partial x_j}\right] + \frac{\varepsilon_f}{k_f}\left(C_1 \nu_f^T S^2 - C_2 \varepsilon_f\right); \tag{2.44}$$

S is the modulus of the mean rate-of-strain tensor expressed as:

$$S = \sqrt{2 S_{ij} S_{ij}}. \tag{2.45}$$

The default coefficients of the $k-\varepsilon$ model are shown in table 2.1.

Table 2.1: Default coefficients of the $k-\varepsilon$ model

Coefficient	C_μ	C_1	C_2	σ_k	σ_ε
Default value	0.09	1.44	1.92	1.0	1.3

2.4.2 Influence of the bubbles on the turbulence (Lahey model)

In a multiphase flow, the bubbles can have some influence on the liquid turbulence as they disturb the flow by creating additional eddies. To account for this, Lahey (2005) proposed a modified $k-\varepsilon$ model. This model adds a source in the transport equation Φ_k and introduces a modified expression for the turbulent viscosity.
The viscosity is expressed as:

$$\nu_f^T = C_\mu \frac{k_f^2}{\varepsilon_f} + 0.6 D \alpha_b |U_r| \tag{2.46}$$

and the equations 2.43 and 2.44 become:

$$\frac{\partial k_f}{\partial t} + \frac{\partial (k_f U_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\nu_f + \frac{\nu_f^T}{\sigma_k}\right)\frac{\partial k_f}{\partial x_j}\right] + \nu_f^T S^2 - \varepsilon_f + \Phi_k \tag{2.47}$$

and

$$\frac{\partial \varepsilon_f}{\partial t} + \frac{\partial (\varepsilon_f U_i)}{\partial x_i} = \frac{\partial}{\partial x_j}\left[\left(\nu_f + \frac{\nu_f^T}{\sigma_\varepsilon}\right)\frac{\partial \varepsilon_f}{\partial x_j}\right] + \frac{\varepsilon_f}{k_f}\left(C_1 \nu_f^T S^2 - C_2\left(\varepsilon_f + \Phi_k\right)\right). \quad (2.48)$$

The expression of the source term is:

$$\Phi_k = C_p\left(1 + C_d^{4/3}\right)\alpha_b \frac{|U_r|}{D}. \quad (2.49)$$

The application of the theory of potential flow around a sphere gives:

$$C_p = 0.25. \quad (2.50)$$

The values of the others modifiable parameters are the same as in the classic $k - \varepsilon$ model displayed in table 2.1.

2.4.3 The turbulent fluctuation velocity

The RANS turbulence models all rely on the Reynolds decomposition of the velocity (already written in equation 2.38):

$$U = \overline{U} + u \quad (2.51)$$

with $\overline{U}$ the mean velocity and u the turbulent fluctuation. In a RANS simulation, only $\overline{U}$ is simulated while it can be necessary to account for the turbulent velocity. It is possible in a Lagrangian simulation to account for this velocity using a stochastic tracking model. One of the models is the gradient dispersion model (Vallier, 2011). In this model, the velocity is perturbed in the direction of $-\vec{\nabla}k$ with a Gaussian random number distribution of variance σ defined as:

$$\sigma = \sqrt{\frac{2k_f}{3}}. \quad (2.52)$$

Thus

$$u = -X\vec{\nabla}k \text{ with } X \sim \mathcal{N}(0, \sigma) \quad (2.53)$$

with $\mathcal{N}$ the gaussian distribution.
This model is very basic and more sophisticated ones exist but the study of stochastic dispersion model is a very broad field whereas its effect on the solutions is not predominant in the study. Therefore, it has not been studied in depth.

2.5 Coalescence and breakage processes

The diameter of the bubble is a crucial parameter that strongly influences the interaction between the two phases. The bubble size distribution is not constant but may change due to bubble-bubble and bubble-turbulent eddies interactions that can lead to breakage and coalescence. However, the mechanisms that drive this process are complex and depend on several factors.

2.5.1 Coalescence and breakage mechanisms

Coalescence

Coalescence is the process by which two bubbles merge during contact to form a single daughter bubble. The process can be described by different consecutive stages: the bubbles collide and a thin film is created between the surface of the two bubbles; this film thickens over a period of time until it reaches a critical thickness and breaks resulting in a single new bubble (Kocamustafaogullari and Ishii, 1995). An example of a coalescence process is shown in figure 2.3.

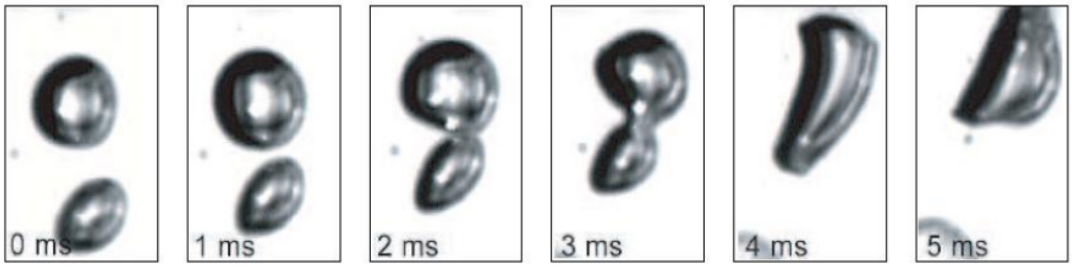


Figure 2.3: Coalescence of two bubbles (extracted from Gharaibah, 2008, fig. 2.10)

Thus this process can be characterized by:

- the frequency of collision f between particles;
- the particle coalescence efficiency η which determines what fraction of fluid particle collision leads to a coalescence event;
- a minimum particle volume V_{min} which is the minimum stable particle size below which a pair of particle will coalesce almost immediately upon colliding;

The main phenomena that drive the collisions between particles are:

- the turbulent fluctuations due to collisions resulting from the random motion of bubbles due to the turbulence of the flow;
- the wake entrainment due to the acceleration of a smaller bubble located in the wake of a bigger preceding bubble;
- the difference of rise velocity between two bubbles with different diameters;
- the shear layer induced velocity difference due to bubbles located in a region of relatively high velocity that may collide with bubble located in a lower velocity region;

Breakage

Breakage of bubbles happens when an external stress exceeds the surface tension stress of the bubbles σ (the force that assures the cohesion of the bubble) (Kocamustafaogullari and Ishii, 1995). This creates some daughter particles that will be more stable as they will have a smaller diameter. Hence, the breakage can also be seen as the collision between a bubble and a turbulent eddy.
The break-up process can be characterized by:

- the maximum particle volume V_{max} which is the maximum stable volume that a bubble can attain in a stagnant flow;

- the daughter particle distribution β;
- the number of daughter particle production n;
- the break-up frequency f;

Usually, only binary break-up are considered: a bubble will break into two daughter particle that will have the same volume. Thus $\beta = \delta_{D/2}$ and $n = 2$.
The main phenomena that create an external stress at the surface of the bubble are:

- the fluctuating eddies present in a turbulent flow which create a pressure variation at the surface of the bubble and hence an additional external stress;
- the viscous shear in the continuous phase in laminar flow;
- the interfacial instability of the bubble (like the Rayleigh-Taylor and Kelvin-Helmhotz instabilities);

2.5.2 Implementation in a multiphase flow formulation

The coalescence and breakage processes can be implemented in both Eulerian-Eulerian and Eulerian-Lagrangian formulation. This is done by introducing a new equation in the two-fluid model (the IAC equation) and by introducing reacting particles in the Lagrangian Particle Tracking. However, only the Eulerian-Eulerian implementation will be studied in depth.

Eulerian-Lagrangian framework

In a Lagrangian formulation, the frequency of collision does not need to be modelled as the occurrence of the collision is natively taken into account by the particle tracking. The rest of the processes can be compared to a chemical reaction and reacting model developed for chemical simulation (like diesel injection) can be used and adapted. This can lead from a simple model (a simple efficiency factor) to a complex one (with a kinetic model and activation energy for example). However, coalescence and break-up in a Lagrangian framework will not be studied in this work and the modelling will only be done in the Eulerian-Eulerian framework.

Eulerian-Eulerian framework

In an Eulerian-Eulerian framework, only the mean diameter of the bubbles can be computed for every cell. To compute it, the equation of conservation of the mass for bubbles of same volume is expressed (in a incompressible formulation) (Ishii et al., 2002)

$$\frac{\partial f(V)}{\partial t} + \nabla \cdot (f(V)U_b) = S_c(V) + S_b(V), \tag{2.54}$$

with $f(V)$ the number of bubbles of volume V, S_c the formation rate of bubbles of volume V per unit volume and S_b the loss rate of particle per unit volume.
This equation is then integrated between V_{min} and V_{max}:

$$\frac{\partial n}{\partial t} + \nabla \cdot (nU_b) = R_c + R_b, \tag{2.55}$$

with n the total number of particles of all sizes per volume, and R_c and R_b the mean formation and loss rate of bubbles.

This equation will be derived to obtain the interfacial area concentration (IAC) equation. The IAC is defined as:

$$a_i = nA_i \tag{2.56}$$

with A_i the average surface area of fluid particles. This can be directly linked to the mean diameter of the bubbles:

$$a_i = \frac{A_b}{V} = \frac{\alpha_b A_b}{V_b} = \frac{6\alpha_b}{D}. \tag{2.57}$$

From equation 2.55, the IAC equation can be deduced :

$$\frac{\partial a_i}{\partial t} + \nabla \cdot (a_i U_b) = \phi_c + \phi_b \tag{2.58}$$

with $\phi = A_i R = \frac{a_i}{n} R$ and $\alpha = nV$, one can find that:

$$\phi_c + \phi_b = \frac{1}{3\psi} \left(\frac{\alpha}{a_i}\right)^2 (R_c + R_b) \text{ with } \psi = \frac{1}{36\pi}. \tag{2.59}$$

The mean diameter can be found for every computational cell by solving equation 2.58. For a better numerical stability at small void fraction, some solvers are solving instead the interfacial curvature equation $\kappa = \frac{a_i}{\alpha} = \frac{6}{D}$ directly derived from equation 2.58.

2.5.3 Models for the formation and loss rate terms

To model the formation and the loss rate of each bubble as introduced in equation 2.54, two models are usually used: Wu et al. (1998) and Hibiki and Ishii (2002). Both of them account for coalescence due to random collision; coalescence due to wake entrainment and bubble breakup rate due to turbulent impact as illustrated in figure 2.4.
All of these are occurring after a collision between two bubbles or between a bubble and a turbulent eddy. Therefore, a general expression of the rate of coalescence and breakage can be defined as:

$$R = \pm f \times \eta \times n \tag{2.60}$$

with f the frequency of the happening of the collision, η its efficiency (ie. the percentage of collision that actually leads to breakup or coalescence) and n the bubble number density ($n = \frac{\alpha}{V}$).

Bubble Coalescence Due to Random Collisions

The coalescence rate due to Random Collision R_{rc} is expressed as:

$$R_{rc} = -f_{rc} \times \eta_{rc} \times n. \tag{2.61}$$

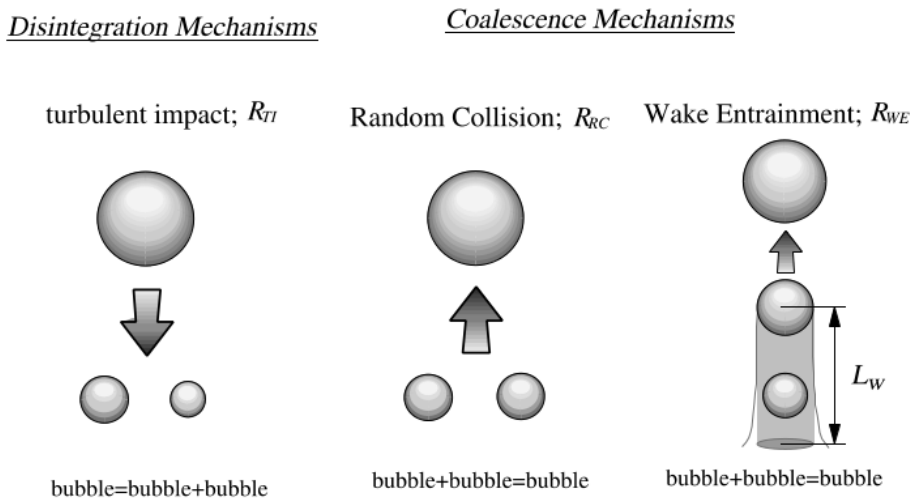


Figure 2.4: Major bubble interaction mechanisms in a bubbly flow (extracted from Ishii et al., 2002, fig.2)

Wu et al. model Wu et al. (1998) proposed to determine f_{rc} by assuming that bubble collision is happening between neighbouring bubbles only. The collision frequency for two bubbles moving toward each other is estimated as well as a correction factor that characterizes the probability that a bubble moves toward a neighbouring bubble. Then another modification factor is suggested to account for the situation when the distance between the bubbles is too large and thus no collision would happen:

$$f_{rc} = \frac{\varepsilon^{1/3} D^{7/3} n}{\alpha_{max}^{1/3}\left(\alpha_{max}^{1/3} - \alpha^{1/3}\right)} \left[1 - \exp\left(-C \frac{\alpha_{max}^{1/3}\alpha^{1/3}}{\alpha_{max}^{1/3} - \alpha^{1/3}}\right)\right]. \tag{2.62}$$

To model the efficiency term of the coalescence, a constant coefficient η_{RC} is chosen. Thus,

$$R_{rc} = \frac{\varepsilon^{1/3} \alpha^2 C_{RC}}{D^{11/3}\alpha_{max}^{1/3}\left(\alpha_{max}^{1/3} - \alpha^{1/3}\right)} \left[1 - \exp\left(-C \frac{\alpha_{max}^{1/3}\alpha^{1/3}}{\alpha_{max}^{1/3} - \alpha^{1/3}}\right)\right]. \tag{2.63}$$

Chen et al. (2005) proposed after extensive testing to take $C = 3$, $\alpha_{max} = 0.8$ and $C_{RC} = 0.021$.

Hibiki and Ishii model Hibiki and Ishii (2002) considered instead that the bubbles are behaving like ideal gas molecules. f_{RC} is then expressed as a function of the surface available for the collision to take place and of the volume available to the collision:

$$f_{rc} = C_{rc} \frac{\alpha \varepsilon^{1/3}}{D^{2/3}(\alpha_{max} - \alpha)}. \tag{2.64}$$

The efficiency factor is not a constant anymore but relies on the assumption that coalescence occurs if the contact time between two bubbles exceeds the time required for the complete film. Thus:

$$\eta_{rc} = \exp\left(-\frac{t_c}{\tau_c}\right) = \exp\left(-K_c \frac{\varepsilon^{1/3}\rho_f^{1/2} D^{5/6}}{\sigma^{1/2}}\right). \tag{2.65}$$

So finally,

$$R_{rc} = C_{rc} \frac{\alpha^2 \varepsilon^{1/3}}{D^{11/3}(\alpha_{max} - \alpha)} \exp\left(-K_c \frac{\varepsilon^{1/3}\rho_f^{1/2} D^{5/6}}{\sigma^{1/2}}\right). \tag{2.66}$$

The different factor K_c, C_{rc} and α_{max} are determined experimentally. Taitel et al. (1980) and Coulaloglou and Tavlarides (1977) proposed that $K_c = 1.29$, $\alpha_{max} = 0.52$ and $C_{RC} = 0.005$.

Bubble Coalescence due to Wake Entrainment

The coalescence rate due to wake entrainment R_{we} is expressed as:

$$R_{we} = -f_{we} \times \eta_{we} \times n. \tag{2.67}$$

This mechanism has only been studied by Wu et al. (1998). f_{we} is calculated by determining the number of bubbles present in the effective volume, in which the following bubbles may collide with the leading one. This volume will depend on the wake region length L_w which is determined experimentally and is usually seen as a constant: $L_w = 7D$ (Tsuchiya et al., 1989). Thus:

$$f_{we} = \frac{7}{8}\pi D^2 U_r n. \tag{2.68}$$

where U_r is the bubble velocity relative to the liquid. Rather than the exact expression of U_r, the relative velocity is roughly estimated from a balance between the buoyancy and the drag force:

$$U_r = \left(\frac{D\left(\rho_f - \rho_b\right) g}{3 C_D \rho_f}\right)^{1/2}. \tag{2.69}$$

The expression of the drag coefficient is developed in section 2.3.1.
The efficiency is treated as constant factor. So finally:

$$R_{we} = C_{we} U_r \frac{\alpha^2}{D^4} \text{ with } C_{we} = 0.0073. \tag{2.70}$$

Bubble Breakup due to Turbulent Impact

The bubble breakup rate caused by turbulent impact is expressed as:

$$R_{ti} = f_{ti} \times \eta_{ti} \times n. \tag{2.71}$$

Wu et al. model Wu et al. (1998) proposed a model depending on a critical Weber number We_{cr}. The Weber number is an nondimensional number already defined in equation 2.3
f_{TI} is expressed as:

$$f_{ti} = \exp\left(-\frac{\mathrm{We}_{cr}}{\mathrm{We}}\right). \tag{2.72}$$

The efficiency η_{ti} is determined by the assumption that bubble breakup caused by turbulent eddies impact occurs when the turbulent eddies have enough energy to overcome the surface tension of the bubble. So for $\mathrm{We} \leq \mathrm{We}_{cr}$, no break-up will occur. And for $\mathrm{We} \geq \mathrm{We}_{cr}$,

$$\eta_{ti} = \frac{u_t}{D}\left(1 - \frac{\mathrm{We}_{cr}}{\mathrm{We}}\right). \tag{2.73}$$

So finally:

$$R_{ti} = C_{ti}\frac{\alpha\varepsilon^{1/3}}{D^{11/3}}\left(1 - \frac{\mathrm{We}_{cr}}{\mathrm{We}}\right)\exp\left(-\frac{\mathrm{We}_{cr}^2}{\mathrm{We}^2}\right). \tag{2.74}$$

The adjustable parameters, $C_{ti} = 0.0945$ and $\mathrm{We}_{cr} = 2$ have been determined experimentally.

Hibiki and Ishii model Hibiki and Ishii (2002) still consider that the bubbles behave like perfect gas. They also make the assumption that only eddies with the same diameter as the bubble will break it, as the larger eddies will transport the bubbles and the smaller won't have the sufficient energy to break the bubble.
f_{ti} is expressed as:

$$f_{ti} = C_{ti}\frac{\alpha\varepsilon^{1/3}}{D^{2/3}(\alpha_{max} - \alpha)}. \tag{2.75}$$

The efficiency is expressed as:

$$\eta_{ti} = \exp\left(-\frac{\overline{E_b}}{\overline{e}}\right) = \exp\left(-K_B\frac{\sigma}{\rho_f D^{5/3}\varepsilon^{2/3}}\right) \tag{2.76}$$

with $\overline{e}$ the average enery of a single eddy and $\overline{E_b}$ the average energy to break the bubble. So finally:

$$R_{ti} = C_{TI}\frac{\alpha(1-\alpha)\varepsilon^{1/3}}{D^{11/3}(\alpha_{max} - \alpha)}\exp\left(-K_B\frac{\sigma}{\rho_f D^{5/3}\varepsilon^{2/3}}\right). \tag{2.77}$$

3

Numerical implementation

The OpenFOAM software is used to numerically implement the problem based on all the models described in section 2. This software is a free and open-source CFD software package written in C++, object-orientated and incorporates numerous multiphase solvers. Two suitable solvers are chosen : one Eulerian-Eulerian solver, `twoPhaseEulerFoam`, and one Lagrangian solver, `DPMFoam`. These two solvers are described in section 3.1 and 3.2. The geometry and boundarty conditions of the problem are then set-up in section 3.3.

3.1 Eulerian-Eulerian implementation

`twoPhaseEulerFoam` is chosen to solve the problem with the Eulerian-Eulerian approach. The solver is described by the OpenFOAM documentation as "a system of two compressible fluid phases with one phase dispersed including heat-transfer"and is based on the procedure described by Rusche (2002, section 3.2).
The version used is based on OpenFOAM 3.0, slightly modified to remove the heat transfer equations.
The solution procedure used by the solver relies on a collocated grid and on the PIMPLE algorithm. This algorithm is a merge between the PISO algorithm (by the construction of a pressure correction equation: $\nabla^2 P = f(\vec{\nabla} P, \vec{U})$) and the SIMPLE algorithm (with the idea of the relaxation of the variables).
To implement this algorithm, two variables (`nCorrector` and `nOuterCorrectors`) must be defined and the relaxation factors have also to be set if necessary. For non orthogonal meshes, an additional correction can also be applied but this is not developped here. Hence, the procedure can be rewritten in pseudo-code in figure 3.1.
The main parts of this algorithm is now described.

3.1.1 The mass conservation equation

First, the mass conservation equation is solved for the dispersed phase (equation 2.10) and a new α_b is found. This is done using a MULES (Multidimensional Universal Limiter with Explicit Solution) solver. The process behind it specifically applied to `twoPhaseEulerFoam` is explicited by Manni (2014). This solving method is used in order to ensure the boundedness of α_b but is a fully explicit solver: the Courant number must be small to have convergence.

for *N from 1 to NOuterCorrectors* **do**
  Solve the mass-conservation equation for the dispersed phase $\rightarrow \alpha_b^{new}$;
  Deduce the void fractions from equation 2.9 $\rightarrow \alpha_f^{new}$;
  Update the coefficients of the interfacial moments term ;
  Construction and discretization of the implicit terms of momentum equation (equation 3.2);
  Relaxation of this equation ;
  for *N2 from 1 to NCorrectors (Pressure correction loop)* **do**
    Prediction of the fluxes from the velocity field $\rightarrow \Phi^{new}$;
    Solve $\nabla^2 P = f(\nabla P, \Phi) \rightarrow P^{new}$;
    Correction of the fluxes with the new pressure gradient ;
    Pressure relaxation ;
    Reconstruction of the velocities from the corrected fluxes $\rightarrow U^{new}$;
  end
  Solve the turbulent equation and update the viscosity term;
end

Figure 3.1: Pseudo code of the solution procedure of `twoPhaseEulerFoam`

α_f is then determined using the equation 2.9:

$$\alpha_f = 1 - \alpha_b. \tag{3.1}$$

Then, the diameter of the bubble is reevaluated by solving the IAC equation (section 2.5.2) and the residuals of the two mass equations (R_b and R_f) are computed.

3.1.2 The momentum equation

The magnitude, linearity and uniformity of the inter-phase momentum transfer term in the momentum equation are known to affect the stability of the solution procedure. Therefore, special attention is given on how to treat these terms.
Hence:

- The drag term is treated semi-implicitly in both the continuous and dispersed phase momentum equation. For the dispersed phase, the part dependent on $\vec{U}_b$ is treated implicitly whereas the part dependent on $\vec{U}_f$ is treated explicitly. The contrary goes for the continuous phase.
- The virtual mass force is treated implicitly.
- Because it is difficult to make an implicit treatment of them, the lift force, the wall lubrication and the turbulent dispersion force are treated explicitly. The turbulent dispersion force is also incorporated in the mass conservation equation as it has a diffusive effect on the phase fraction distribution.

The terms that are treated implicitly correspond to the equation:

$$\frac{\partial}{\partial t}\left(\alpha_b \rho_b \vec{U}_b\right) + \alpha_b \rho_b \left(\vec{U}_b \cdot \vec{\nabla}\right) \vec{U}_b = \alpha_b \left(\nabla \cdot \nu \vec{\nabla}\right) \vec{U}_b + \vec{M}_{VM} - K\vec{U}_b. \tag{3.2}$$

These equations are relaxed before the drag is added and can be rewritten as:

$$A_b\vec{U}_b = H_b \tag{3.3}$$

and

$$A_f\vec{U}_f = H_f. \tag{3.4}$$

The momentum equations 2.11 and 2.13 are now discretized by interpolating the velocities at the cell faces. Using the notation of equations 3.4 and 3.3, the volumetric phase equations are written:

$$\phi_b = \frac{H_b}{A_b} + \phi_b^M - \frac{\alpha_b}{A_b}\vec{\nabla}P_\phi + \alpha_b\frac{K}{A_b}\phi_f \tag{3.5}$$

and

$$\phi_f = \frac{H_f}{A_f} + \phi_f^M - \frac{\alpha_f}{A_f}\vec{\nabla}P_\phi + \alpha_f\frac{K}{A_f}\phi_b \tag{3.6}$$

with ϕ^M the terms treated explicitly except the drag (ie. wall lubrication, lift, turbulent dispersion and gravity fluxes) and P_ϕ the flux of the pressure.
One should note that these equations and the following ones are discrete equations, therefore the different operators ($\nabla\cdot$ or $\vec{\nabla}$) are only used to indicate that a specific discretization scheme is used.
These equations are rewritten

$$\phi_b = \Phi_b - \Gamma_b\vec{\nabla}P_\phi \tag{3.7}$$

and

$$\phi_f = \Phi_f - \Gamma_f\vec{\nabla}P_\phi. \tag{3.8}$$

By combining the equations 3.5 and 3.6, one can note that $\phi_r = \phi_b - \phi_f$ can be expressed without ϕ_b and ϕ_f:

$$\phi_r = \frac{\left(\phi_b^s + K_b^d\phi_f^s\right) - \left(\phi_f^s + K_f^d\phi_b^s\right)}{1 - K_b^dK_f^d} \tag{3.9}$$

with $K_f^d = \alpha_f\frac{K}{A_f}$, $K_b^d = \alpha_b\frac{K}{A_b}$, $\phi_b^s = \frac{H_f}{A_f} + \phi_f^M - \frac{\alpha_f}{A_f}\vec{\nabla}P_\phi$ and $\phi_f^s = \frac{H_f}{A_f} + \phi_b^M - \frac{\alpha_b}{A_b}\vec{\nabla}P_\phi$.

3.1.3 The pressure correction equation

In fact, the flux equations are never directly solved. Instead, a pressure correction equation is derived from the mass-conservation and the momentum equation. Thus it has been introduced:

$$U = U^* + U' \tag{3.10}$$

and

$$P_\phi = P_\phi^* + P_\phi' \tag{3.11}$$

with the $*$ subscript referring to the old value and the $'$ subscript referring to the correction term.
By combining the two mass-conservation equations (equations 2.10 and 2.12), one can find that

$$\nabla\cdot\vec{U} = 0 \text{ with } \vec{U} = \alpha_b\vec{U}_b + \alpha_f\vec{U}_f. \tag{3.12}$$

Therefore:

$$\nabla \cdot \vec{U}' = -\nabla \cdot \vec{U}^* = -R_f - R_b. \tag{3.13}$$

This equation is discretized and as the equations 3.5 and 3.6 are still valid for ϕ', it becomes:

$$\nabla \cdot \Phi - \nabla^2 \left(\Gamma P'_\phi\right) = -\frac{R_f}{\rho_f} + \frac{R_b}{\rho_b} \tag{3.14}$$

with $\Phi = \alpha_b \Phi_b + \alpha_f \Phi_f$ and $\Gamma = \alpha_b \Gamma_b + \alpha_f \Gamma_f$.
Then, the fluxes are corrected and a new global flux can be found:

$$\phi = \Phi + \Gamma \vec{\nabla} P_\phi. \tag{3.15}$$

And ϕ_b and ϕ_f are deduced using ϕ_r: (which is recomputed using the new gradient pressure)

$$\phi_b = \phi + \alpha_f \phi_r \tag{3.16}$$

and

$$\phi_f = \phi - \alpha_b \phi_r. \tag{3.17}$$

Finally the velocities are reconstructed from the corrected fluxes in order to avoid oscillations that may occur on collocated grids, the $k - \varepsilon$ equations are solved and the turbulent viscosity is updated.

3.2 Eulerian-Lagrangian implementation

`DPMFoam` and `MPPICFoam` are the solvers chosen to solve the problem with the Eulerian-Lagrangian approach. According to the OpenFOAM documentation, it is "a transient solver for the coupled transport of a single kinematic particle cloud including the effect of the volume fraction of particles on the continuous phase and the collision between particles". `MPPICFoam` is based on the same code than `DPMFoam` but without the collisions between particles which is taken into account with an additional force as presented in section 2.3.6.
The version used is based on OpenFOAM 3.0, slightly modified to make it compatible with a buoyant fluid.The solving process can be rewritten in pseudo-code as shown in figure 3.2.
One can see that the mass-conservation equation of the continuous phase is not solved but deduced from the evolution of the cloud. This ensures a better stability compared to the Eulerian-Eulerian implementation where the mass-conservation equation is difficult to solve.

3.2.1 Equations in the Eulerian and Lagrangian frame

Equations in the Lagrangian frame

For each particle, equation 2.17 is applied and integrated over an Eulerian timestep Δt. A new velocity is found from the previous velocities and the forces acting on the particles:

$$\vec{U}(t + \Delta t) = \frac{\Delta t}{m_b} \left(\vec{M}_b(t) + \rho \vec{g}\right) + \vec{U}(t). \tag{3.18}$$

for *every particle P* **do**
    Look for possible collision;
    Divide the Eulerian time-step into sub Lagrangian time-step ;
    Derive the new velocities and position of each particle ;
    Compute the influence of the particles on the fluid $\vec{M}_f^{(P)}$;
end
Compute the new void fraction α_b^{new} from the position of each particle ;
Deduce the void fractions from equation 2.9 $\rightarrow \alpha_f^{new}$;
for *N2 from 1 to NCorrectors (Pressure correction loop)* **do**
    Prediction of the fluxes from the velocity field $\rightarrow \Phi^{new}$;
    Solve $\nabla^2 P = f(\nabla P, \Phi) \rightarrow P^{new}$;
    Correction of the fluxes with the new pressure gradient ;
    Pressure relaxation ;
    Reconstruction of the velocities from the corrected fluxes $\rightarrow U^{new}$;
end
Solve the turbulent equation and update the viscosity term;

Figure 3.2: Pseudo code of the solution procedure of `DPMFoam` and `MPPICFoam`

The new positions of the particles are then easily evaluated. The process to account for collisions is described in section 3.2.2.
In practice, a particle trajectory can cross several cells during an Eulerian time step Δt. This is why Δt is divided into a set of Lagrangian time steps, specific for each particle to account for the time it enters and/or leaves a computational cell.

Equations in the Eulerian frame

Once the new position of each particle has been determined, the void fraction α_b and α_f can be computed. The momentum equation of the continuous phase (equation 2.16) is solved based on the PIMPLE algorithm.
$\vec{M}_f$ (which represents the interaction of the particles with the fluid) is evaluated based on the difference of the particle momentum between the two timesteps. For example, for a given cell A and a given particle P.

- If the particle is present in the cell A at the instant t and is still there at the instant $t + \Delta t$. Its contribution $\vec{M}_f^{(P)}$ on the cell A is expressed by:

$$\vec{M}_f^{(P)} = \frac{m_b}{V \Delta t}(\vec{U}(t + \Delta t) - \vec{U}(t)). \tag{3.19}$$

- If the particle is in the cell A at the instant t but leaves it during the timestep at the point F, the instant when the particle will leave the cell is estimated (t') and a new time step $\Delta t' = t' - t$ is defined. The contribution of the particle is:

$$\vec{M}_f^{(P)} = \frac{m_b}{V \Delta t'} \left[\vec{U}(t + \Delta t') - \vec{U}(t)\right]. \tag{3.20}$$

By counting for every cell the contribution of all particles present during whole or part of the timestep, $\vec{M}_f$ can be determined for each computation cell. This term is

treated implicitly in the Navier-Stokes equations which can thus be written as

$$A_f \vec{U}_f = H_f. \tag{3.21}$$

As the fluid is considered incompressible, the divergence of the velocity is null and a pressure correction equation ($\nabla^2 P = f(\nabla P, \Phi)$) can be developped and from that a new velocity field for the fluid can be found based on the same method than the one presented in section 3.1.3

3.2.2 Implementation of the collision

The collision between particles are also taken into account in `DPMFoam`. Let's consider a particle P_i (at the position $\vec{x}_i$ and with a velocity $\vec{U}_i$) and its possible collision with the particle P_j (at the position $\vec{x}_j$ and with a velocity $\vec{U}_j$). A local coordinate is set centred on the particle with $\vec{n}_{i\to j}$ the normal vector which goes from particle i to particle j and $\vec{t}_{i\to j}$ the tangential vector. The velocities can be defined as:

$$\vec{U}_i = U_i^N \vec{n}_{i\to j} + U_j^T \vec{t}_{i\to j} \tag{3.22}$$

and

$$\vec{U}_j = U_j^N \vec{n}_{i\to j} + U_j^T \vec{t}_{i\to j} \tag{3.23}$$

A collision occurs if:

- Their trajectory intersect. That is to say if: $\left(U_i^N - U_j^N\right) \vec{n}_{i\to j} \geq 0$.
- Their relative displacement is larger than the distance between them (including their diameter): $\left(U_i^N - U_j^N\right) \Delta t \leq |\vec{x}_i - \vec{x}_j| - D$.

A hard sphere model is implemented with the introduction of a coefficient of restitution e (which quantifies the loss of energy during a collision) but a more sophisticated model can also be implemented (for example a spring-slider model). It is assumed that the tangential velocity will not change during the collision so the new velocities $U_i^{N'}$ and $U_j^{N'}$ can be deduced with the conservation of the energy and a given coefficient of restitution:

$$\frac{1}{2} m_b \left(U_i^{N'}\right)^2 + \frac{1}{2} m_b \left(U_j^{N'}\right)^2 = \frac{1}{2} m_b \left(U_i^N\right)^2 + \frac{1}{2} m_b \left(U_j^N\right)^2 \tag{3.24}$$

and

$$e = \frac{U_j^{N'} - U_i^{N'}}{U_j^N - U_i^N}. \tag{3.25}$$

The collision with a wall relies on the same type of process.

3.3 Implementation of the problem

A two-phase simulation can then be carried on by the two previous solvers. It remains now to define the input of these solvers.

3.3.1 Geometry and physical properties

The dimension of the plate is based on the one used in a similar project ongoing at the university. The plate has a length of 1.6m and a width of 1m. All the simulations will be done in 2D and thus the width will not have any influence. The flow is studied 0.2m below the plate: as the bubbles are naturally getting close to the wall, the flow becomes quickly unaffected by the bubbles with the depth. The physical properties of the two phases are constant and a given rate flow of air $\eta\dot{Q}$ (with $\eta \in [0, 1]$) is injected in the system. The value of these parameters are displayed in table 3.1. The diameter of the bubbles is not determined and the influence of the diameter on the solution will be studied in the following but it is considered that the diameter will be between 0.1mm and 10mm. From the values of the table, the Morton number can be computed $\text{Mo} = 2.8 \times 10^{-14}$ and proves that the water is a very low Morton number fluid.

Table 3.1: Physical properties of the phases

Property	Value
ρ_f	1000 kg.m^{-3}
ρ_b	1.25 kg.m^{-3}
ν_f	1×10^{-6} m^2.s^{-1}
ν_b	1.3×10^{-5} kg.m^{-3}
σ	0.7 N.m^{-1}
$\dot{Q}$	0.00132 m^3.s^{-1}
D	0.1mm - 10 mm

The location of the ejection of the bubbles outside the air cavity is really difficult to determine and would require a very long study. Hence, it has been decided to create a patch measuring 5cm in the vicinity of the wall where the bubbles are injected with a constant distribution. All the air bubbles are in their final longitudinal velocity (2m/s). As only the steady state is studied and as the inlet of the bubbles aren't right, 1m is added to let the time to the flow to be in steady state. The 2-D geometry is displayed in figure 3.3

3.3.2 Boundary conditions

The boundary conditions must also be set carefully as it influences the accuracy of the final results and the stability of the solver. The boundary conditions are the same for the two formulations (Eulerian-Eulerian and Eulerian-Lagrangian) except for the quantities α_b and U_b which do not exist in the Lagrangian solver. The patch name are defined as in figure 3.3. The coefficient of restitution are set to $e = 0.97$ for a particle colliding a wall as well as for a collision between two particles.

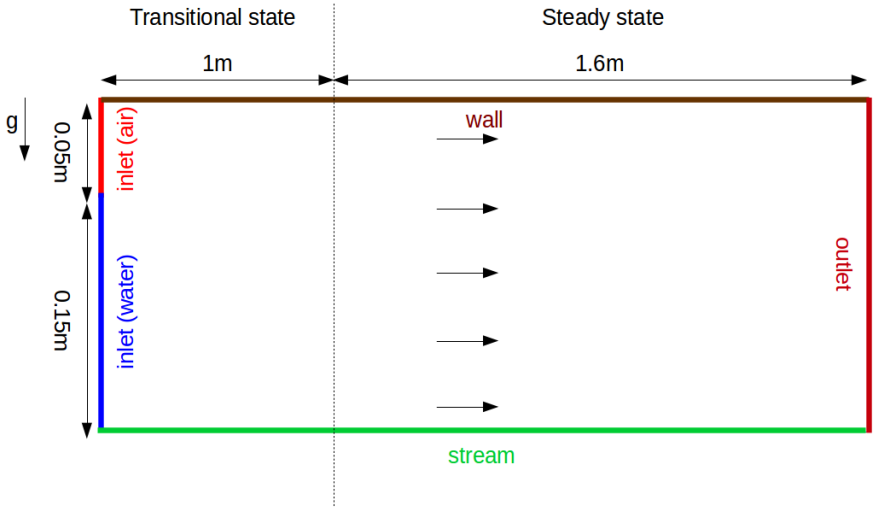


Figure 3.3: 2-D geometry of the problem

Void fraction and number of particles

The void fraction at the inlet(air) is found based on the definition of the volume air flow $\dot{Q}$. At the patch inlet (water), the void fraction is, of course, set to $\alpha_b = 0$.

$$\alpha_b = \frac{\dot{Q}}{U_b A} = 10\dot{Q}. \tag{3.26}$$

In a Lagrangian formulation, the number of particles introduced per second in the system N is found based on the air flow rate:

$$\dot{Q} = NV = N\frac{\pi}{6}D^3 \tag{3.27}$$

However, in the case of a 2D study, the number of particle must be lowered as in this solver, all the particles are injected in the centre of every mesh. As there is just one mesh in the width direction in a 2D study, the number of particle injected will be too high. The domain actually studied has just a width of the diameter of the bubble. Thus:

$$N^{(2D)} = \frac{6\dot{Q}}{\pi D^2}. \tag{3.28}$$

Velocities and pressure

The velocities of the fluid and the bubbles are set to 2m/s at the inlet. At the wall, a slip condition – which sets the normal component to the wall of the gas velocity equal to zero – is chosen for the bubble phase whereas the velocity of the fluid phase is set to zero as no-slip is expected. For the outlet an InletOutlet condition is set: this condition is normally a zeroGradient condition but can switch to a fixedValue if a back-flow occurs.
Regarding the pressure, the boundary conditions are set for the quantity p_{rgh} which is the pressure without hydrostastic pressure.

$$p_{rgh} = p - \rho g h \text{ with } h \text{ the depth of the fluid.} \tag{3.29}$$

A constant fixed value equal to the atmospheric pressure is set at the outlet whereas a `fixedFluxPressure` is set at the inlet so that the flux on the boundary is the one specified by the velocities boundary condition.

Turbulent quantities

For the wall, wall functions are used as boundaries conditions for both of the turbulent quantities. For the inlet, the turbulent quantities are estimated as:

$$k_f = (0.05U_f^2) = 0.01\text{m}^2.\text{s}^{-2} \tag{3.30}$$

and

$$\varepsilon_f = 0.54\frac{k^{3/2}}{0.1h} = 0.027\text{m}^2.\text{s}^{-3} \text{ with } h \text{ the height of the inlet.} \tag{3.31}$$

The summary of all the boundary conditions is displayed in table 3.2 and 3.3.

Table 3.2: Boundary conditions for the inlets and outlet

Field	inlet (air)	inlet (water)	outlet
U_b	fixedValue	fixedValue	inletOutlet
U_f	fixedValue	fixedValue	inletOutlet
α_b	fixedValue	fixedValue	inletOutlet
ε_f	fixedValue	fixedValue	inletOutlet
k_f	fixedValue	fixedValue	inletOutlet
κ (IAC)	fixedValue	fixedValue	zeroGradient
ν_f^T	calculated	calculated	calculated
p	calculated	calculated	calculated
p_{rgh}	fixedFluxPressure	fixedFluxPressure	fixedValue (10^5 Pa)

Table 3.3: Boundary conditions for the wall and stream patches

Field	wall	stream
U_b	slip	zeroGradient
U_f	fixedValue (0m/s)	zeroGradient
α_b	zeroGradient	zeroGradient
ε_f	epsilonWallFunction	zeroGradient
k_f	kqRWallFunction	zeroGradient
κ	zeroGradient	zeroGradient
ν_f^T	calculated	calculated
p	calculated	calculated
p_{rgh}	zeroGradient	zeroGradient

3.3.3 Meshing, discretization scheme and closure models

It is assumed that the shape of the bubble will be spherical everywhere. Therefore, based on section 2.3, the models for the closure terms are selected for spherical

bubbles and are displayed in table 3.4. After the simulation, it will be check if Ta ≤ 1 as stated in section 2.1.2.

Table 3.4: Choice of the models for the closure interface

Force	Model	Solver
Drag	Schiller-Neumann	Lagrangian & Eulerian
Virtual force	Constant ($C_{VM} = 0.5$)	Lagrangian & Eulerian
Lift	Constant ($C_L = 0.5$)	Lagrangian & Eulerian
Wall lubrication	Frank	Eulerian
Turbulent diffusion	Lopez - De Bortano	Eulerian
Collision	Snider	Lagrangian (MPPIC)
Aspect ratio	$\phi = 1$	Lagrangian & Eulerian

The choice of the discretization schemes are based on the proposal from Michta (2011) and on the schemes used in the tutorial *fluidizedBeds* provided with OpenFOAM 3.0. The same schemes are used for LPT and Euler-Euler. The scheme's choice is displayed in table 3.5. The relaxation factor applied to the velocity of the two phases is set to 0.4. The value of `nOuterCorrectors` is set to 5 and the one of `nCorrectors` is set to 2. A residual control for the PIMPLE loop is also introduced to reduce the computation time if the solution converges quickly.

Table 3.5: Choice of the discretization schemes

Name of the scheme	Scheme
ddtSchemes	Euler
laplacianSchemes	Gauss linear uncorrected
gradSchemes	Gauss Linear
divSchemes	Gauss upwind
interpolationSchemes	Linear
snGradSchemes	uncorrected

A simple mesh is used for the problem. The main objectives is to have orthogonal meshes (`twoPhaseEulerFoam` being very sensitive to that) and to have a mesh fine enough to see the void fraction distribution close to the wall but which does not require a too small timestep (required to ensure a small Courant number). It has been chosen to use a fine mesh close to the wall (from 0 to 0.05m) with cells of dimension 1mmx5mm and a coarse mesh for deeper flow with a cells of dimension 10mmx5mm. This gives 21000 cells in total. The mesh is shown in figure 3.4. It is checked for each simulation that the first node is located in the log-law region (ie. $30 \leq y^+ \leq 100$)

3.3.4 Post-processing of the results

The shear force present because of the friction between the fluid and the plate creates a resistance force for the body called the viscous resistance. As it can easily

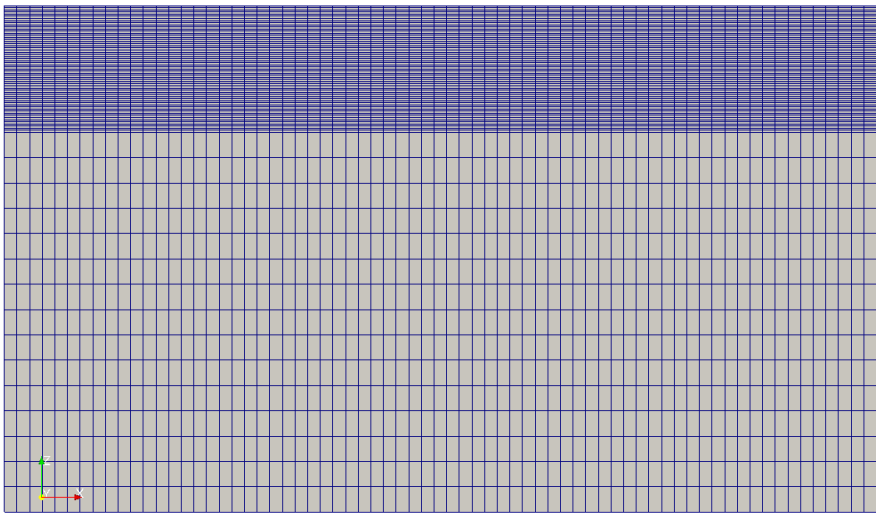


Figure 3.4: The mesh used for the study

be assumed that only the friction between the water and the body will create this force (ie. the contribution of the friction between the air and the plate is neglected) the viscous resistance is expressed as:

$$F = (1 - \alpha_b)\tau_w \tag{3.32}$$

with τ_w the local shear stress defined as:

$$\tau_w = \mu_f \left(\frac{\partial u}{\partial y}\right)_{y=0} \tag{3.33}$$

with u the flow velocity parallel to the wall and y the near-wall distance. In order to easily compare the skin-friction between different geometry or velocity of the flow, the C_F, coefficient skin fraction is used and defined as:

$$C_F = \frac{\tau_w}{0.5\rho U_\infty^2}. \tag{3.34}$$

So finally:

$$C_F = \frac{\nu_f + \nu_f^T}{0.5U_\infty^2} \left(\frac{\partial u}{\partial y}\right)_{y=0}. \tag{3.35}$$

In OpenFOAM, the gradient of U is computed with the utility `wallGradU` slightly modified to read and compute the field `U.water` instead of `U`. From that the skin friction coefficient is easily calculated using equation 3.35.

4

Results and discussions

By using the mathematical formulation and the solvers available in OpenFOAM described in sections 2 and 3, the case presented in section 3.3 can finally be simulated. The major axis of study will be the reduction of the viscous resistance experienced by the plate (ie. the value of the skin friction factor) and the comparison of the results given by the solvers between the two main formulations (Euler-Euler and Euler-Lagrangian).

4.1 General results

The first simulations are done using the Eulerian-Eulerian solver. As for every other simulations, the consistency of the output of a solver must be checked. If no experimental datas have been found for this precise case, the shape and behaviour of the flow can be checked qualitatively with similar experiments. A comparison with the situation without bubbles is also done to study and estimate the influence of the bubbles on the flow.

4.1.1 Behaviour of a bubbly flow

A first simulation is done using the `twoPhaseEulerFoam` solver. The properties of the simulation are displayed in table 4.1. One can note that the $k-\varepsilon$ model is used and not the Lahey model while this model considers the turbulence created by the bubbles and hence is said to be more accurate. This is because the Lahey model is not implemented in the Lagrangian solvers and, as one of the goal is to compare this output with the other solvers, the more identical the input are, the better it is.

Table 4.1: Properties of the Eulerian-Eulerian simulation

Property	Value
Solver	`twoPhaseEulerFoam`
α (inlet)	0.0132
Diameter	1mm
Turbulence model	$k-\varepsilon$
Timestep	Variable (set so that Co = 0.5)

The distribution of α_b is presented in figure 4.1. The bubbles are concentrated very close to the wall and the influence of the bubbles becomes quickly negligible with the

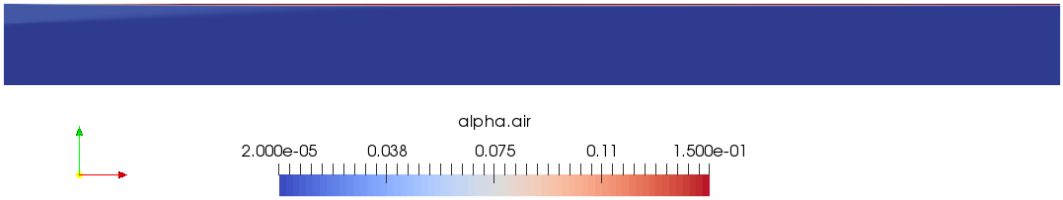
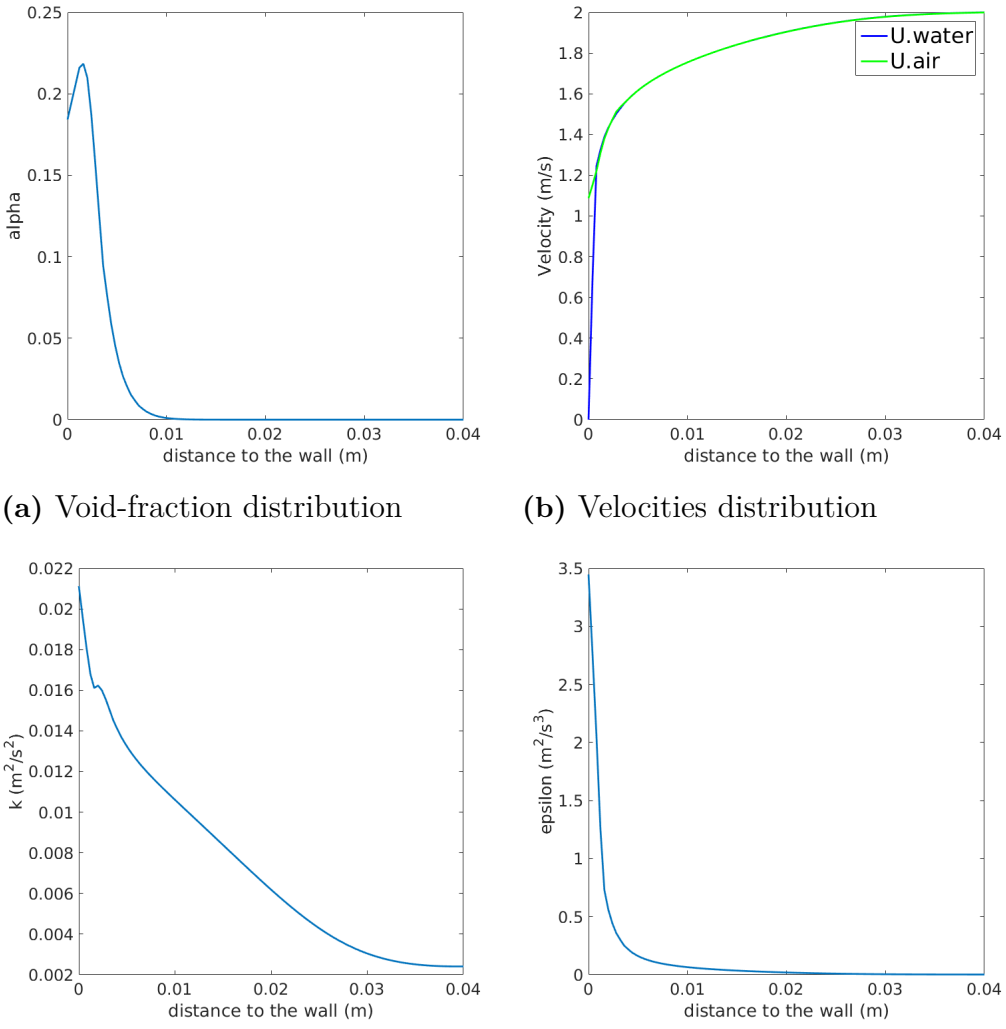


Figure 4.1: Distribution of the void fraction over the domain



(a) Void-fraction distribution

(b) Velocities distribution

(c) Turbulent kinetic energy k distribution

(d) Turbulent dissipation ε distribution

Figure 4.2: Distribution of the quantities wrt. the distance to the wall

depth. This bubble pattern may be related to the fact that when the bubble reaches a steady state, the predominant force over the bubble motion is the buoyant force. It is noted that after 0.9m, the different quantities are not dependent anymore on the distance to the inlet. Thus, the flow is steady after 1m, as assumed in figure

3.3. The flow rate of the air through this section is also computed by integrating the quantity $\alpha \times U_b$ over the section and is equal to $\dot{Q}$ as expected. To have a better idea of the behaviour of the flow close to the wall, the lateral profile of the different quantities (at steady state) are displayed in figure 4.2.
The void fraction (figure 4.2a) has a peak distribution as described in the litterature and its shape is similar to comparable experiments like the one from Yoshida et al. (1998, fig.5).
Regarding the velocities (figure 4.2b), one can see that, although the velocity profiles of the bubble and fluid phase are very similar in the streamwise direction, there are some slight differences between them. The bubble velocity is higher than the fluid velocity because the bubbles are free of the restriction of the wall no-slip boundary condition but becomes very similar far from the wall. The velocity distribution is very similar to the one found for a similar experiment (Pang et al., 2014, fig. 6a).
The hypothesis that the bubbles are in the spherical regime is checked by plotting the Ta number (defined in section 2.1.2). This is shown in figure 4.3. It can be seen that the Tadaki number is largely below 1 everywhere in the domain. So, it can safely be assumed now that the bubbles are in spherical regime.

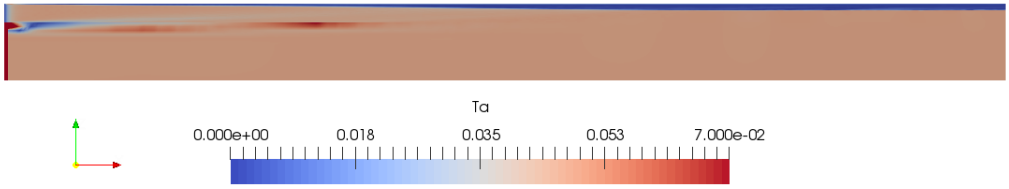


Figure 4.3: Spatial distribution of the Tadaki number

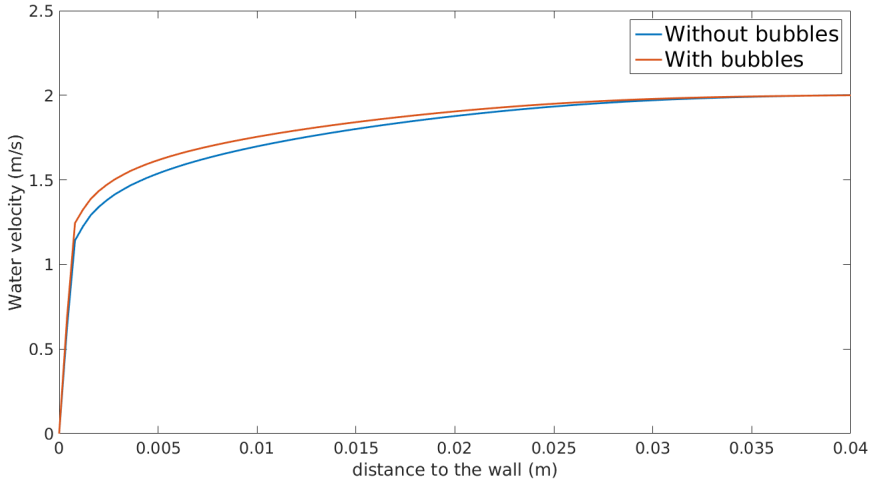
4.1.2 Behaviour without injection of bubbles

A simulation of a case without injection of any bubbles is done and its parameters are shown in table 4.2. This allows to estimate the influence of the bubbles on the fluid phase.

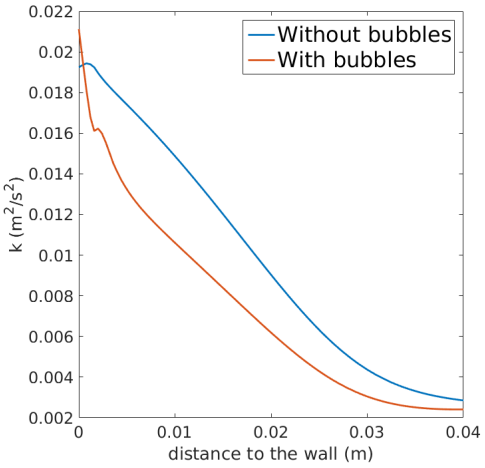
Table 4.2: Properties of the simulation without injection of bubbles

Property	Value
Solver	`twoPhaseEulerFoam`
α (inlet)	0
Diameter	1mm
Turbulence model	$k - \varepsilon$
Timestep	Variable (set so that Co = 0.5)

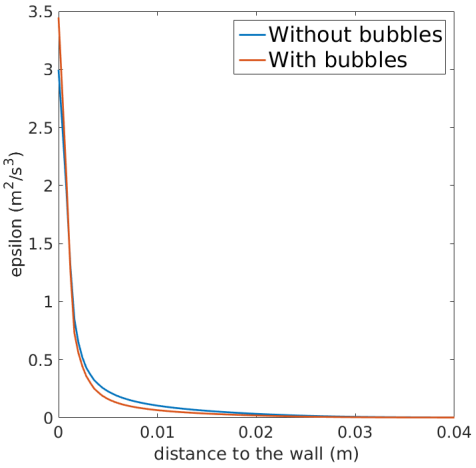
The lateral profile of the quantities is shown figure 4.4 and is compared to the results found in the Eulerian-Eulerian simulation done in section 4.1.1.
Regarding the velocity (figure 4.4a), one can note that the velocity in presence of bubbles is slightly enhanced in the region away from the wall. Regarding the



(a) Velocities distribution



(b) Turbulent kinetic energy k distribution



(c) Turbulent dissipation ε distribution

Figure 4.4: Comparison of the quantities with and without bubbles

turbulent quantities (figure 4.4b and 4.4c), only small differences can be observed for ε whereas the shape of k is quite different between the two cases even if the numerical values are still similar.
The skin friction coefficient of the plate is also computed from the results and compared with the one found in section 4.1.1. This is displayed in table 4.3. It can be seen that the presence of the bubbles causes a reduction of the drag of 10 %. If it seems to prove that the friction resistance is reduced by the presence of bubbles, the consistency of this result will be studied more precisely in section 4.3.

Table 4.3: Skin friction coefficient with and without bubbles

	With bubbles	Without bubbles
C_F	2.66×10^{-3}	3.01×10^{-3}

4.1.3 Comparison between the solvers

The results from the Eulerian-Eulerian simulation are then compared to the two Lagrangian solvers `DPMFoam` and `MPPICFoam`. The parameters of the Lagrangian simulations are displayed in table 4.4 and are set so that the cases are as identical as possible to the Eulerian one. It can be noted that the time needed to run the simulation are similar for the three solvers.

Table 4.4: Properties of the LPT simulations

Property	Simulation LPT	Simulation MPPIC
Solver	`DPMFoam`	`MPPICFoam`
Parcels per second	2521	2521
Diameter	1mm	1mm
Turbulence model	$k-\varepsilon$	$k-\varepsilon$
Timestep	0.002	Variable (set so that Co = 0.5)

The output of `DPMFoam` is shown in figure 4.5.

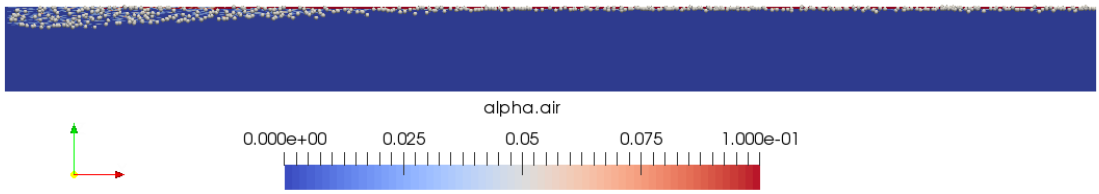
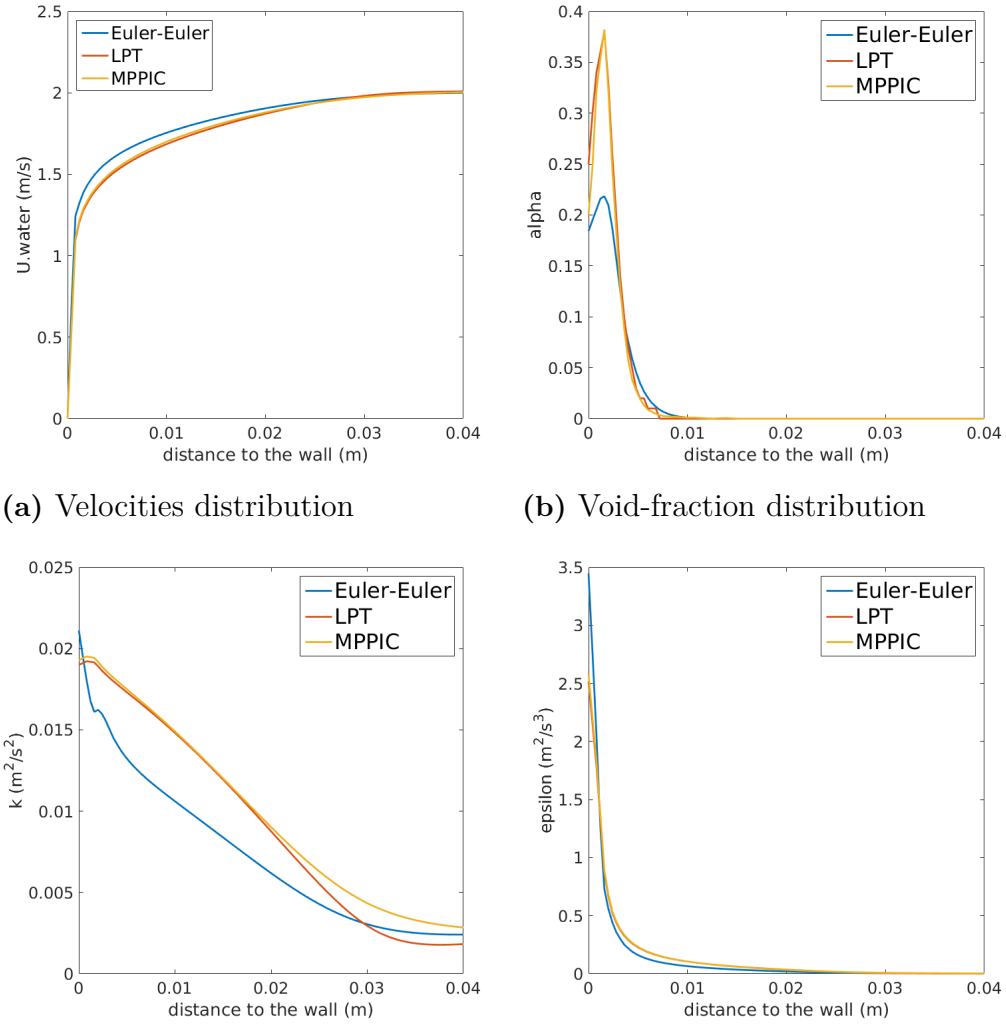


Figure 4.5: Typical output of a Lagrangian solver

The quantities are now compared between every solver and their behaviour close to the wall is shown in figure 4.6. In overall, it can be seen that both the MPPIC and LPT give almost exactly the same results for this case.
Regarding the velocity of the water (fig. 4.6a), the velocity field found from the Lagrangian solvers is slightly smaller than the one found with the Eulerian solver. This leads to an increase of the rate of the flow of bubbles towards the wall for the Lagrangian solver and therefore will certainly increase the void fraction at the wall. This is confirmed by the void fraction distribution (fig. 4.6b) that shows that the values of void fraction at the wall (which has a strong influence on the friction resistance) is smaller for the Eulerian-Eulerian solver. However the shape of the two curves are very similar. The peak is higher in the Lagrangian frame but located at the same distance to the wall.
Regarding the turbulent quantities (fig. 4.6c and 4.6d), ε is very similar for the three solvers but the shape of k differs slightly. It can be noted that the shape of k found with the Lagrangian solvers is very similar to the one found in the case without bubbles (figure 4.4b).
The skin friction coefficient C_F is also computed for the three methods and compared to see if the differences in the lateral profile have a macroscopic effect. The friction coefficients are shown in table 4.5.



(a) Velocities distribution

(b) Void-fraction distribution

(c) Turbulent kinetic energy k distribution

(d) Turbulent dissipation ε distribution

Figure 4.6: Comparison of the quantities between the tree solvers wrt. the distance to the wall

Table 4.5: Comparison of the friction coefficients

	`twoPhaseEulerFoam`	`DPMFoam`	`MPPICFoam`	Without bubbles
C_F	2.66×10^{-3}	2.14×10^{-3}	2.30×10^{-3}	3.01×10^{-3}

It can be seen that the three solvers give quite different results. This is mainly due to differences in the value of the void fraction at the wall as it can be seen figure 4.6b. Such a difference can be problematic, must be taken into account and studied more thoroughly and will be done in section 4.3. The Lagrangian Particle Tracking method is considered to be a simulation at a smaller scale and thus to be more accurate than the two-fluid model. Therefore, it seems that the Eulerian-Eulerian method, while faster to run for 3D simulation, is underestimating the result for this

case.
The simulations done with `MPPICFoam` have not been satisfying. Indeed the time needed to track the particles are quite similar for both `DPMFoam` and `MPPICFoam`. However as for `twoPhaseEulerFoam`, the Euler-Euler part of the MPPIC method requires a very small Courant number to be stable. Therefore, `MPPICFoam` is not really stable and does not gain any computation time: it will not be used in the following.

4.2 Sensitivity of the results regarding the different parameters

The sensitivity of the output of the solver to the choice of some models are then studied in order to check if some of the assumptions made previously are acceptable. The influence of the turbulence model will be studied to check if it can be justified to just use the classic $k - \varepsilon$ model for the Lagrangian simulations. Then, it is also checked if breakage and coalescence happen for this case in order to know if a non-reacting cloud (which only simulate bubbles with a constant diameter over time) can be used with the Lagrangian solver.

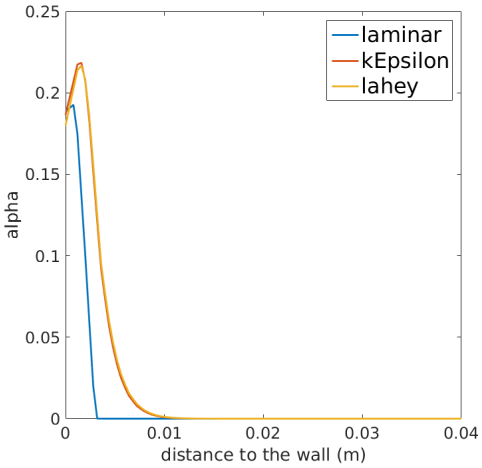
4.2.1 Influence of the turbulence model

Three Eulerian-Eulerian simulations are launched with the same inputs but with three different turbulence models. The properties of the three simulation are displayed in table 4.6 and the lateral profile is displayed in figure 4.7.

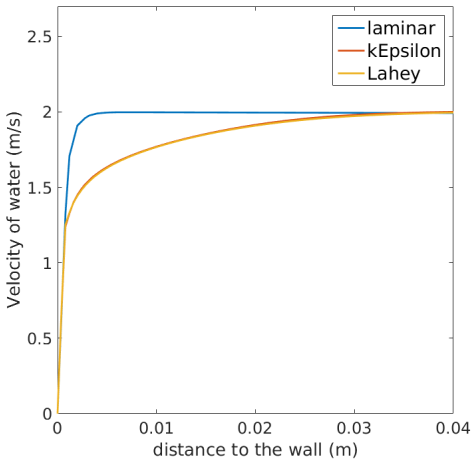
Table 4.6: Properties of the turbulent simulations

Property	laminar	kEpsilon	Lahey
Solver	Euler-Euler	Euler-Euler	Euler-Euler
α(inlet)	0.0132	0.0132	0.0132
Diameter	1mm	1mm	1mm
Turbulence model	laminar	$k - \varepsilon$	Lahey $k - \varepsilon$
Timestep	Variable	Variable	Variable

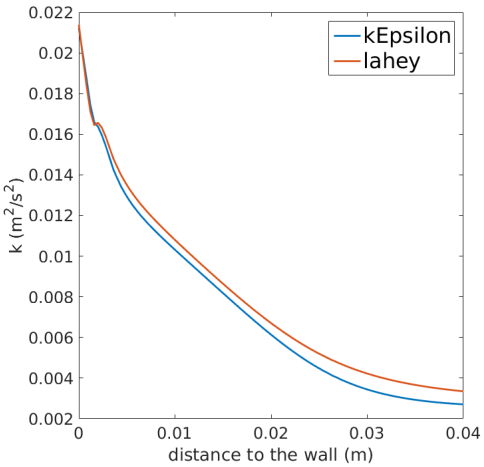
The laminar and turbulent simulations give very different results. The thickness of the boundary layer with the turbulent model is ten times higher than the thickness of the laminar one. The contrary would be surprising as the flow has quite a high Reynolds number (Re $\approx 10^6$) and therefore is highly turbulent.
It can be seen that the bubbles do not influence so much the turbulence quantities for this magnitude of bubble diameter. ε is identical for the two turbulence model but the values of k differ slightly but are almost identical close to the wall. No apparent change in the velocity field is observed. This leads to an almost identical value of skin friction coefficient. This is mainly due to the fact that the bubble concentration is not so high in this case and therefore does not influence so much the turbulence of the flow.



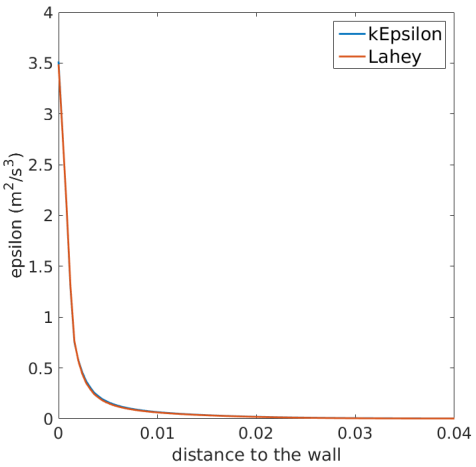
(a) Void-fraction distribution



(b) Velocities distribution



(c) Turbulent kinetic energy k distribution



(d) Turbulent dissipation ε distribution

Figure 4.7: Comparison of the different quantities between solvers

Hence, the choice of the turbulence model is quite free for this case. The classic $k-\varepsilon$ model can safely be used with the Lagrangian simulations. However, as the Lahey model does not cause additional instabilities and that some studies can be done with a denser flow (where the bubbles are more likely to influence the turbulence), the Lahey model is preferred for the Eulerian simulations.

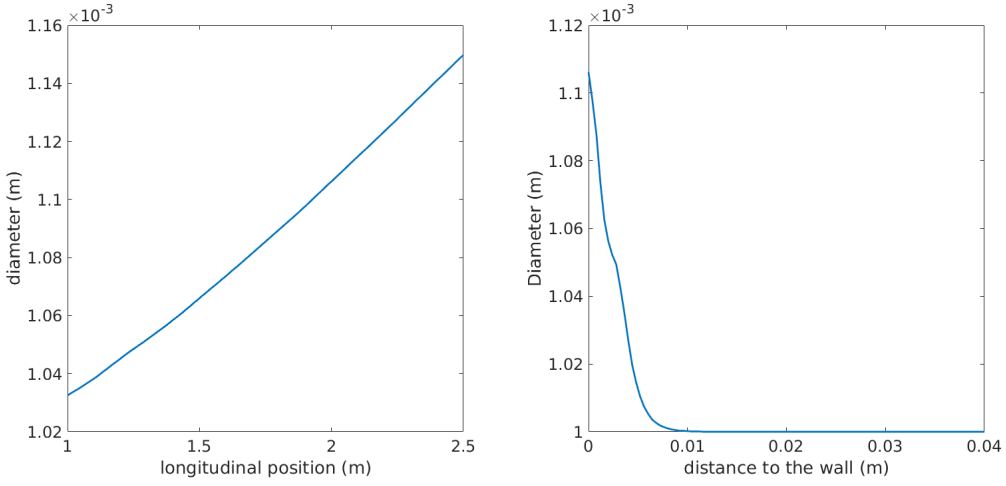
4.2.2 Influence of breaking/coalescence process

In order to study the evolution of the diameter of the bubbles along the plate, an Eulerian-Eulerian simulation is launched where the diameter of the bubble is governed by the IAC equation presented in section 2.5.2. The Wu et al. (1998) model is used to model the breakage and coalescence terms. The properties of the IAC simulation are shown in table 4.7.

Table 4.7: Properties of the IAC simulation

Property	Value
Solver	`twoPhaseEulerFoam`
α (inlet)	0.0132
Diameter	variable, 1mm at the inlet
Turbulence model	Lahey
Timestep	Variable (set so that Co = 0.5)

The diameter of the bubble does not change except very close to the wall. The diameter of the bubble at the wall is progressively increasing with the length as seen in figure 4.8a. The lateral profile (figure 4.8b) shows that the diameter quickly decreases towards its initial value of 1mm with the distance to the wall. One can just consider the diameter of the bubble at the void-fraction peak (as seen in figure 4.2a). This gives a diameter of 1.04mm which can be easily rounded down to the initial diameter of 1mm. The skin friction coefficient is exactly the same than the one found without the IAC equation. Therefore, it can be fairly considered to use a constant diameter and to not solve the IAC equation or implement collision and coalescence in Lagrangian Particle Tracking. If the result is not sensitive to the coalescence and breakage for this case, this hypothesis must be checked again if the geometry is changed to a longer plate, a more turbulent flow (higher velocity for example) or a denser solution.



(a) Diameter of the bubbles at the wall

(b) Lateral distribution of the diameter wrt. the distance to the wall

Figure 4.8: Average diameter of the bubble

4.3 Influence of the diameter of the bubble on the solution

The diameter of the bubble is a crucial parameter that influences the solution as it influences almost every forces that affects the bubbles. Therefore the gain of efficiency must be studied with respect to the diameter of the bubble.

4.3.1 Influence of the diameter in a Lagrangian formulation

Numerous simulations are launched with bubbles' diameter ranging from 0.1mm to 10mm. The study is made using the Lagrangian solver as it is said to give more accurate results than the Eulerian one. The comparison with the Eulerian solver will be done in section 4.3.2.
The duration of the calculation is really dependant on the diameter: for 10mm, only 25 particles are injected per second whereas for 0.1mm, 252 100 particles need to be injected to keep the same air rate of flow. Moreover, when the number of particles increases, the occurrence of collisions is following the same trend which increases the number of Lagrangian timesteps for each eulerian timestep. Especially, the 0.1mm and 0.2mm cases have been really long to compute. For 5.0 and 10.0mm, the small number of particles makes more difficult to render time-averaged curve, the problem is thus simulated for 12s instead of 8s and more timesteps are written. For every simulation, the skin friction coefficient is computed (as described in section 3.3.4. The gain of efficiency is computed as the relative error between the skin friction coefficient of the simulation and the one found without bubbles (presented in table 4.3). The results are displayed in figure 4.9

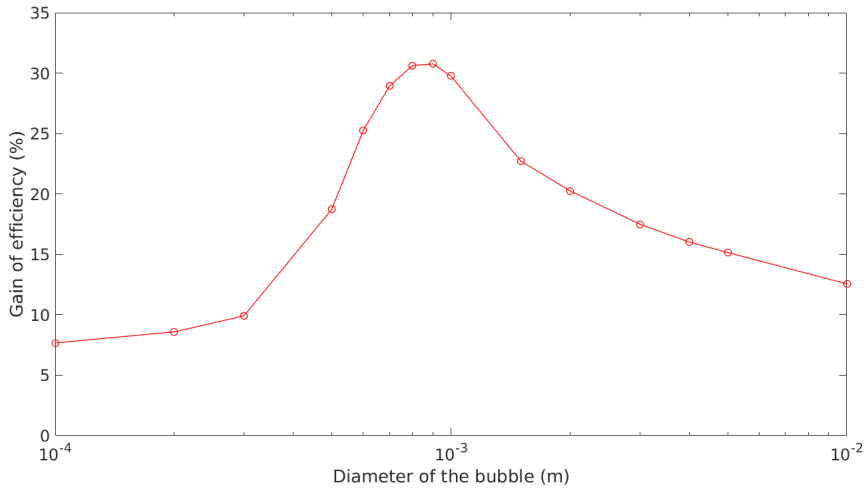


Figure 4.9: Gain of efficiency wrt. diameter with the LPT solver

The skin friction coefficient changes a lot with the diameter of the bubbles. It is observed that the velocity and turbulence fields are almost the same for all diameter (as seen in figure 4.11): this terms are not sensitive to the diameter. The difference in skin friction coefficient is caused by the void fraction's value at the wall as the

void fraction α is really sensitive to the diameter. Three main regimes have been distinguished that have a distinctive shape of void fraction distribution and is shown figure 4.10

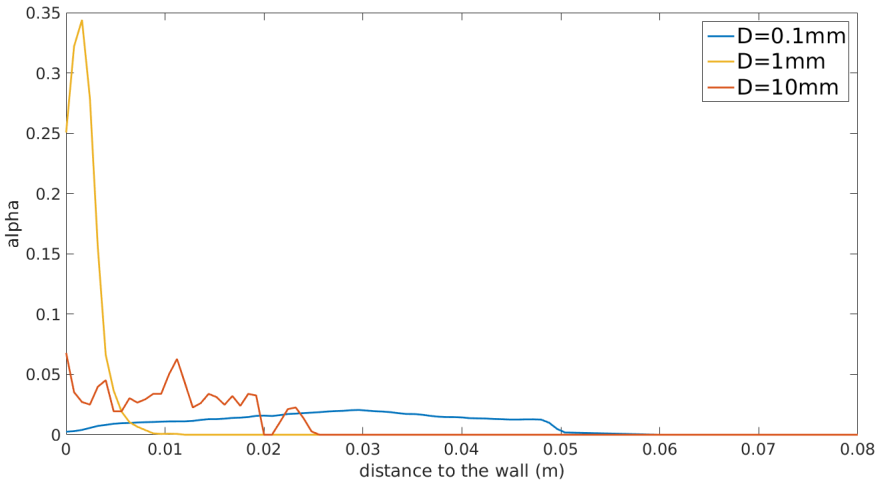
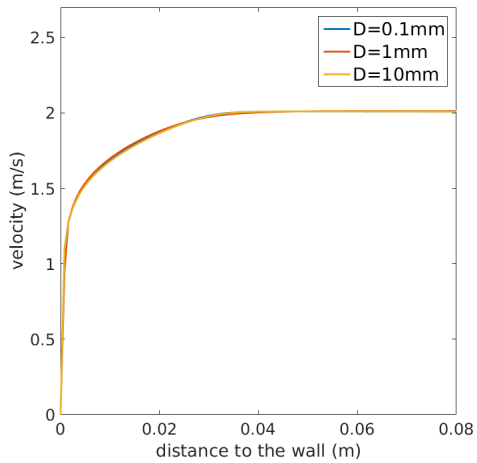


Figure 4.10: Void fraction distribution for different diameters

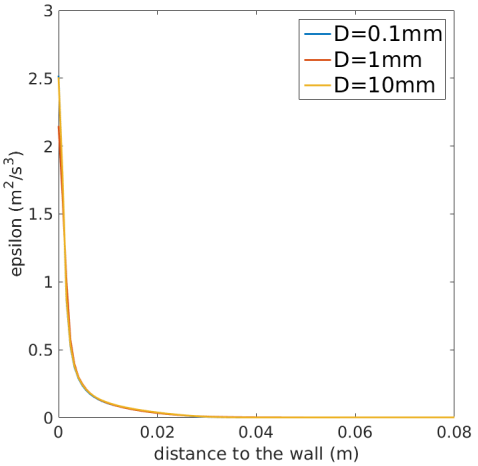
These regimes are:

- A collision-driven distribution for low diameter's bubbles ($D \leq 0.5$mm). To keep the same air rate flow with a small bubble diameter, numerous particle are present and collisions occur very frequently. Therefore, the position becomes really dependent on the collision history (interaction bubbles-bubbles) and less on the forces caused by the bubble-fluid interactions. This leads to a bigger variety of position of the bubbles related to the wall. The peak observed in figure 4.1 no longer exists. Thus, the void fraction at the wall remains small, the gain of efficiency is small (5 %) and is not too dependant on the diameter (slow increase with the diameter).
- A "fluid velocity"-driven distribution for medium diameter's bubble (0.5mm$\leq$ $D \leq$ 3mm). In this regime, the void fraction is not driven so much by the collisions anymore. A peak distribution is found with almost all the bubbles located very close to the wall. This is explained because the bubble still have a quite small diameter and therefore a small inertia. When the bubble, is approaching the wall (driven by the gravity and lift forces) and rebound at the wall, the bubble will be quickly affected by low-velocity region of the water and therefore will stagnate there. This lead to a peak distribution and to an important bubble concentration at the wall that will improve a lot the gain of efficiency. That is why the best efficiency is found for this regime for a diameter where both the effect of the collision and the inertia are negligible.
- An inertia-driven regime for bubbles with a big diameter ($D \geq 3mm$). As the bubble will have a bigger volume and therefore a bigger inertia, the bubbles will not react immediately to the low-velocity field of the water close to the wall. Hence, depending on the velocity and direction of the bubble before the collision with the wall, the bubble stays in the low-velocity zone or comes back to a zone closer to the wall. Therefore, the distribution is not a peak anymore

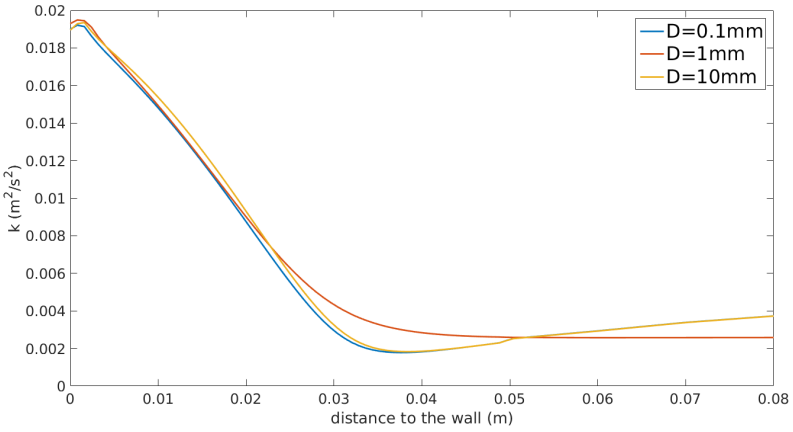
but have a plateau which is dependant on the inertia of the bubble. Therefore, the void fraction at the wall decreases slowly with the diameter and so does the gain of efficiency.



(a) Velocities distribution



(b) Turbulent dissipation ε distribution



(c) Turbulent kinetic energy k distribution

Figure 4.11: Comparison of the quantities wrt. the diametre of the bubbles

One can note that the limit between the collision and the fluid-velocity driven regime is only function of the concentration of particles whereas the limit between the fluid-velocity and inertia regimes are just a function of the mass of the particles (and hence its diameter).

4.3.2 Comparison with the Eulerian formulation

A series of simulations is also launched with the Eulerian-Eulerian solver to compare the results of efficiency with the one found in section 4.3.1. Unlike with the Lagrangian solver, the computation time is not really dependent on the diameter of the bubble as the discrete phase is considered as a continuous one. For an undeter-

mined reason, the solver becomes highly unstable for bubbles with a 3mm diameter so that the simulation for this diameter has not been successfully simulated. The reason of that remains unknown.

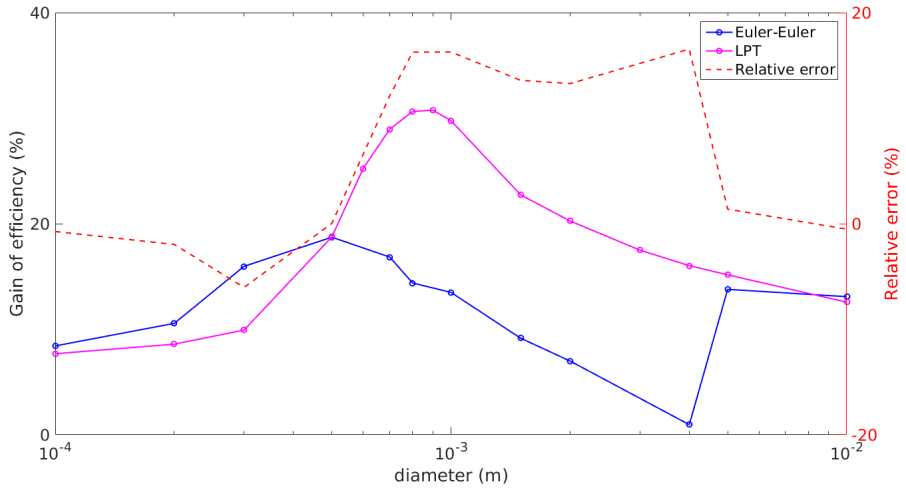
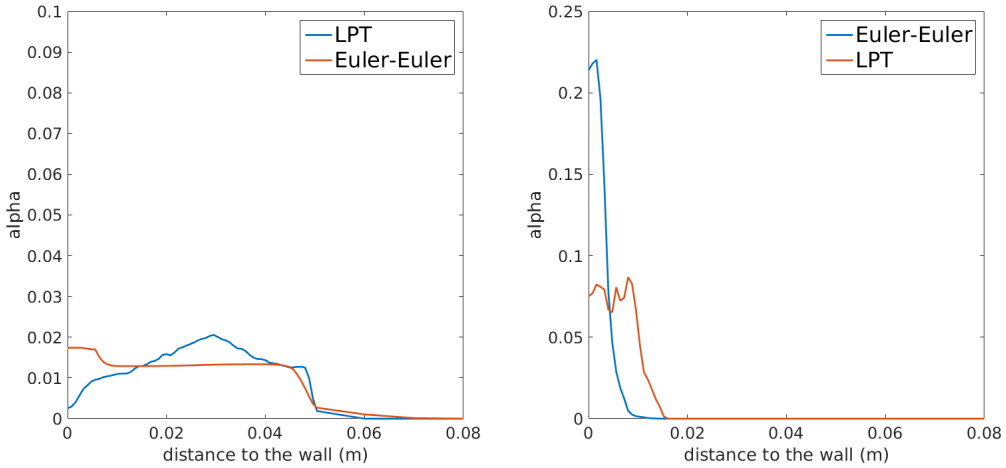


Figure 4.12: Comparison of the skin friction computed by the two solvers

The results are shown in figure 4.12. It can be seen that the results differ greatly from the ones found with the Lagrangian solver. Depending on the diameter, the relative error between the two solvers is between 0 % and 25 %. The reason of these discreptancies are studied considering the regimes defined in section 4.3.1 and by comparing the void fraction distribution.

- For small diameters (the "collision regime"), the results given by the two solvers are similar. The comparison of the void fraction distribution shows an identical shape but the result slightly diverges when close to the wall. This still give a satisfying results as the relative error between the two solvers is lower than 5%. Counting that the computation-time is very long for the 2D problem (and hence even more for a 3D problem), it can be really interesting to use the Eulerian solver for this range of diameter.
- For medium diameter (the "velocity driven regime"), the error is around 15%. The comparison already made in figure 4.6b shows that the height of the peak is underestimated. This may be improved by adjusting the parameters of the wall-lubrication force. For example, the damping parameter could be decreased to have a result closer to the Lagrangian framework.
- For big diameter, a strange behaviour appears as there is a sudden jump in the results at 5mm. The two frameworks give the same results with an error less than 5% but a closer look at the void fraction distribution and the other quantities shows that the lateral profiles given by the two solvers are really different. The similarity may hence be just a coincidence and the Eulerian solver should not be used for this region. This conclusion is understandable, the number of particles for these diameter is so low that having time-averaged curve takes long with the Lagrangian solver. It is therefore difficult to consider the air phase as a continuum for such small concentrations.



(a) Small diameter (D=0.1mm) **(b)** Big diameter (D=10mm)

Figure 4.13: Lateral void-fraction for different diameters

4.3.3 Influence of the rate flow

The evolution of the skin friction coefficient with respect to the air rate flow is also studied. All the results are presented in percentage as function of the initial air flow $\dot{Q} = 0.00132\text{m}^3\text{s}^{-1}$.
For 1mm diameter bubbles, the rate flow shows a linear curve for both the methods between 0 % and 100% rate of flow for bubbles. This is shown in figure 4.14. However, the gain of efficiency for higher rates of flow will certainly attain an asymptote when it will enter the collision-driven regime.

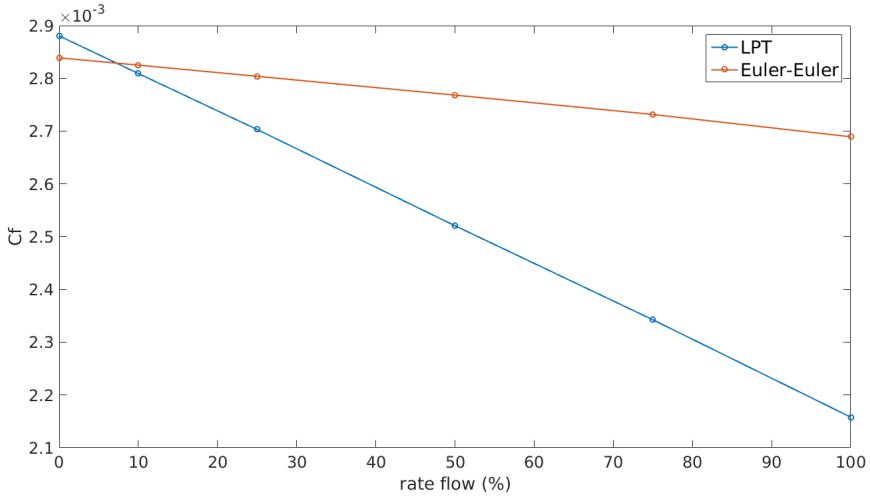


Figure 4.14: Skin friction coefficient wrt. rate flow for 1mm bubbles

Thus, a study for bubble with 0.2mm is also done to study the behaviour of bubbles that are in the collision regime. This is shown in figure 4.15. In this regime, the efficiency curve is almost constant: the rate flow does not influence the gain of efficiency (indeed, more particles would just extend the distribution laterally but

would not increase the void fraction at the wall). One can see that for small rates of flow (from 0 to 10%) a linear curve can be observed which shows that for this range of rates of flow, the bubbles are in the "fluid-velocity regime".

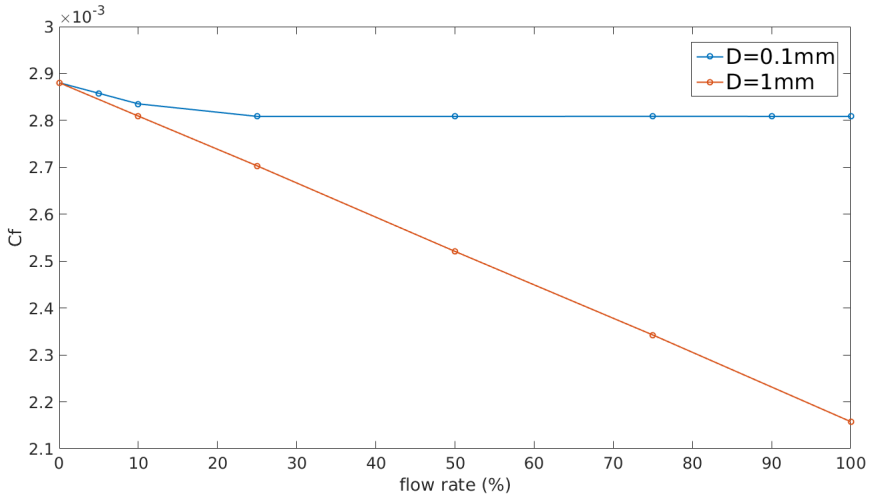


Figure 4.15: Skin friction coefficient wrt. rate flow for different diameter of bubbles

5
Conclusion

This master's thesis gave a first approach on the analysis of a bubble flow under a semi-infinite plate and gave some guidance and recommendations to set-up an efficient method to simulate this flow.

The main result of this study is that an Eulerian-Eulerian solver must be used carefully for this type of analysis as it can lead to very different results than from the Lagrangian solvers. However, for a very dense concentration of bubbles in the boundary layer – when a Lagrangian simulation becomes very computational intensive – the Euler-Euler method can give correct results and save a lot of time compared to the Lagrangian method. For a less dense solution, the Laplacian Particle Tracking solver should always be preferred.

The Eulerian method is not so accurate because it involves a contradiction about the choice of a correct mesh grid: the mesh's size must be big enough to incorporate numerous particles in order to solve average equations but must be small enough to render the behaviour close to the wall (necessary to compute the shear stress at the wall). Moreover, the wall lubrication component's model does not seem suitable for ship simulations: this model has originally been developed empirically to comply with experimental results in a vertical pipe and its generalization to all type of geometry is not convincing. A parametric study or even a new formulation of the wall-lubrication force should be developed to better comply with the reality and can allow the use of an Eulerian-Eulerian solver for a wider range of situations.

It has also been shown that the bubbles are spherical and not subjected to coalescence and breakage. The turbulence model does not influence so much the results and no particular effort should be invested in the choice of it. A first attempt to classify the behaviour of the air flow depending on the concentration of the particles and their volume has been developed.

A three-dimensional study should be done as it would facilitate the comparison with the experimental results and also lead to a better prediction of the real gain of efficiency. In particular, numerous bubbles can escape the area below the plate by the sides and therefore would reduce the expected resistance reduction.

Bibliography

Bhaga, D. and Weber, M. E. (1981). Bubbles in viscous liquids: shapes, wakes and velocities. *Journal of Fluid Mechanics*, 105:61–85.

Burns, A. D., Frank, T., Hamill, I., and Shi, J.-M. (2004). The favre averaged drag model for turbulent dispersion in eulerian multi-phase flows. In *5th International Conference on Multiphase Flow*.

Ceccio, S. and Simo, A. M. (2011). *Air Lubrication Drag Reduction on Great Lakes Ships*. University of Michigan study for Great Lakes Maritime Institute.

Chen, P., Sanyal, J., and Duduković, M. (2005). Numerical simulation of bubble columns flows: effect of different breakup and coalescence closures. *Chemical Engineering Science*, 60(4):1085–1101.

Coulaloglou, C. A. and Tavlarides, L. L. (1977). Description of interaction processes in agitated liquid-liquid dispersions. *Chemical Engineering Science*, 32(11):1289–1297.

Elghobashi, S. (1991). Particle-laden turbulent flows: direct simulation and closure models. *Applied Scientific Research*, 48(3):301–314.

Fan, L.-S. and Tsuchiya, K. (1990). Chapter 2 - single bubble rise characteristics. In *Bubble Wake Dynamics in Liquids and Liquid–Solid Suspensions*, pages 17–69. Butterworth-Heinemann.

Frank, T., Zwart, P. J., Krepper, E., Prasser, H. M., and Lucas, D. (2008). Validation of CFD models for mono and polydisperse air-water two-phase flows in pipes. *Nuclear Engineering and Design*, 238(3):647–659.

Gharaibah, E. (2008). *Entwicklung und Validierung eines Modells polydisperser Zweiphasenströmungen unter Berücksichtigung von Koaleszenz und Dispersion*. Phd thesis, Universität München.

Hibiki, T. and Ishii, M. (2002). Development of one-group interfacial area transport equation in bubbly flow systems. *International Journal of Heat and Mass Transfer*, 45(11):2351–2372.

Ishii, M., Kim, S., and Uhle, J. (2002). Interfacial area transport equation: model development and benchmark experiments. *International Journal of Heat and Mass Transfer*, 45(15):3111–3123.

Ishii, M. and Zuber, N. (1979). Drag coefficient and relative velocity in bubbly, droplet or particulate flows. *AIChE Journal*, 25(5):843–855.

Kawabuchi, M., Mizokami, S., Kodan, Y., Kawakita, C., and Higasa, S. (2011). CFD Predictions of Bubbly Flow around an Energy-saving Ship with Mitsubishi Air Lubrication System. *Mitsubishi Heavy Industries Technical Review*, 48(1).

Kocamustafaogullari, G. and Ishii, M. (1995). Foundation of the interfacial area transport equation and its closure relations. *International Journal of Heat and Mass Transfer*, 38(3):481–493.

Lahey, R. T. (2005). The simulation of multidimensional multiphase flows. *Nuclear Engineering and Design*, 235(10):1043–1060.

Lamb, H. (1932). *Hydrodynamics*. Cambridge Univ. Press, 6.ed., repr edition.

Lopez de Bertodano, M. A. (1998). Two fluid model for two-phase turbulent jets. *Nuclear Engineering and Design*, 179(1):65–74.

Manni, A. (2014). An introduction to twoPhaseEulerFoam with addition of an heat exchange model. Project work for the course in cfd with opensource software, Chalmers University of Technology.

Maritech (2011). The efficacy of air-bubble lubrication for decreasing friction resistance. *Maritech news*, (128).

Michta, E. (2011). Modeling of subcooled nucleate boiling with OpenFoam. Master's thesis, Royal Institute of Technology, Stockholm.

Nygren, A. (2014). *Simulation of bubbly flow in a flat bubble column*. Phd thesis, Lund University.

Pang, M. J., Wei, J. J., and Yu, B. (2014). Numerical study on modulation of microbubbles on turbulence frictional drag in a horizontal channel. *Ocean Engineering*, 81:58–68.

Patankar, N. A. and Joseph, D. D. (2001). Lagrangian numerical simulation of particulate flows. *International Journal of Multiphase Flow*, 27(10):1685–1706.

Rusche, H. (2002). *Computational fluid dynamics of dispersed two-phase flows at high-phase fractions*. Phd thesis, Imperial College, London.

Sanders, W. C., Winkel, E. S., Dowling, D. R., Perlin, M., and Cecci, S. L. (2006). Bubble friction drag reduction in a high-reynolds-number flat-plate turbulent boundary layer. *Journal of Fluid Mechanics*, 552:353–380.

Schiller, L. and Naumann, Z. (1935). A Drag Coefficient Correlation. *VDI Zeitung*, 77:318–320.

Snider, D. M. (2001). An incompressible three-dimensional multiphase particle-in-cell model for dense particle flows. *Journal of Computational Physics*, 170(2):523–549.

Tadaki, T. and Maeda, S. (1961). On the shape and velocity of single air bubbles rising in various liquids. *Chemical engineering*, 25(4):254–264.

Taitel, Y., Bornea, D., and Dukler, A. E. (1980). Modelling flow pattern transitions for steady upward gas-liquid flow in vertical tubes. *AIChE Journal*, 26(3):345–354.

Tomiyama, A., Tamai, H., Zun, I., and Hosokawa, S. (2002). Transverse migration of single bubbles in simple shear flows. *Chemical Engineering Science*, 57(11):1849–1858.

Tsuchiya, K., Miyahara, T., and Fan, L. S. (1989). Visualization of bubble-wake interactions for a stream of bubbles in a two-dimensional liquid-solid fluidized bed. *International Journal of Multiphase Flow*, 15(1):35–49.

Vakhrushev, I. A. and Efremov, G. I. (1970). Interpolation formula for computing the velocities of single gas bubbles in liquids. *Chemistry and Technology of Fuels and Oils*, 6(5):376–379.

Vallier, A. (2011). *Eulerian and lagrangian cavitation related simulations using OpenFOAM*. Thesis for the degree of licenciate of engineering., Lund University.

Wu, Q., Kim, S., Ishii, M., and Beus, S. G. (1998). One-group interfacial area transport in vertical bubbly flow. *International Journal of Heat and Mass Transfer*, 41(8-9):1103–1112.

Yoshida, Y., Takahashi, Y., Kato, H., and Guin, M. M. (1998). Distribution of void fraction in bubbly flow through a horizontal channel: Bubbly boundary layer flow, 2nd report. *Journal of Marine Science and Technology*, 3(1):30–36.

A

Definition of the geometry and the boundary conditions

A.1 Files used by the two solvers

A.1.1 system/blockMeshDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      blockMeshDict;
}
convertToMeters 1;

vertices
(
  (0 0.5 0)
  (2.6 0.5 0)
  (2.6 -0.5 0)
  (0 -0.5 0)

  (0 0.5 -0.05)
  (2.6 0.5 -0.05)
  (2.6 -0.5 -0.05)
  (0 -0.5 -0.05)

  (0 0.5 -0.20)
  (2.6 0.5 -0.20)
  (2.6 -0.5 -0.20)
  (0 -0.5 -0.20)
);

blocks
(
  hex (0 1 2 3 4 5 6 7) (520 1 50)
      simpleGrading (1 1 1)
   hex (4 5 6 7 8 9 10 11) (520 1 15)
       simpleGrading (1 1 1)
);

edges
(
);

patches
(
  patch inlet
  (
    (0 3 7 4)
  )

  patch inlet_water
  (
      (4 7 11 8)
  )

  patch outlet
  (
      (2 1 5 6)
      (6 5 9 10)
  )

  wall walls
  (
      (3 2 1 0)
  )

  patch stream
  (
      (11 10 9 8)
  )
);

mergePatchPairs
(
);
```

A.1.2 0/epsilon.water

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      epsilon.water;
}

dimensions      [0 2 -3 0 0 0 0];

internalField   uniform 0.027;

boundaryField
{
  inlet
  {
    type              fixedValue;
    value             uniform 0.027;
  }

  inlet_water
  {
    type              fixedValue;
    value             uniform 0.027;
  }

  outlet
  {
    type              inletOutlet;
    phi               phi.water;
    inletValue        $internalField;
    value             $internalField;
  }

  walls
  {
    type
        epsilonWallFunction;
    value             $internalField;
  }

  stream
  {
    type              zeroGradient;
  }
}
```

A.1.3 0/k.water

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      k.water;
}

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform  0.01;

boundaryField
{
  inlet
  {
    type              fixedValue;
    value             $internalField;
  }

  inlet_water
  {
    type              fixedValue;
    value             $internalField;
  }

  outlet
  {
    type              inletOutlet;
    phi               phi.water;
    inletValue        $internalField;
    value             $internalField;
  }

  walls
  {
    type              kqRWallFunction;
    value             $internalField;
  }

  stream
  {
    type              zeroGradient;
  }
}
```

A.1.4 0/nut.water

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      nut.water;
}

dimensions      [0 2 -1 0 0 0 0];

internalField   uniform 1e-8;

boundaryField
{
  inlet_water
  {
    type              calculated;
    value             $internalField;
  }

  inlet
  {
    type              calculated;
    value             $internalField;
  }

  outlet
  {
    type              calculated;
    value             $internalField;
  }

  walls
  {
    type              nutkWallFunction;
    value             $internalField;
    }

  stream
  {
    type              calculated;
    value             $internalField;
  }
}
```

A.1.5 0/U.water

```
FoamFile
{
    version     2.0;
    format      binary;
    class       volVectorField;
    object      U.water;
}
dimensions      [0 1 -1 0 0 0 0];

internalField   uniform (2 0 0);

boundaryField
{
  inlet
  {
    type              fixedValue;
    value             uniform (2 0 0);
  }

  inlet_water
  {
    type              fixedValue;
    value             uniform (2 0 0);
  }

  outlet
  {
    type           inletOutlet;
    phi             phi.water;
    inletValue      uniform (0 0 0);
    value           uniform (0 0 0);
  }

  walls
  {
    type              fixedValue;
    value             uniform (0 0 0);
  }

  stream
  {
    type              zeroGradient;
  }
}
```

A.1.6 constant/g

```
FoamFile
{
    version     2.0;
    format      ascii;
    class
        uniformDimensionedVectorField;
    location    "constant";
    object      g;
```

```
}

dimensions      [0 1 -2 0 0 0 0];
value           (0 0 -9.81);
```

A.2 Specific files for twoPhaseEulerFoam

A.2.1 0/alpha.air

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    location    "0";
    object      alpha.air;
}

dimensions      [0 0 0 0 0 0 0];

internalField   uniform 0;

boundaryField
{
  inlet
  {
    type              fixedValue;
    value             uniform 0.0132;
  }

  inlet_water
  {
    type              fixedValue;
    value             uniform 0;
    }

  outlet
  {
    type              inletOutlet;
    phi               phi.air;
    inletValue        uniform 1;
    value             uniform 1;
  }

  walls
  {
    type              zeroGradient;
  }

  stream
  {
    type              zeroGradient;
  }
}
```

A.2.2 0/p

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}

dimensions          [1 -1 -2 0 0 0 0];

internalField       uniform 1e5;

boundaryField
{
  inlet
  {
    type              calculated;
    value             $internalField;
    }

  inlet_water
  {
    type              calculated;
    value             $internalField;
  }

  outlet
  {
    type              calculated;
    value             $internalField;
  }

  walls
  {
    type              calculated;
    value             $internalField;
  }

  stream
  {
    type              calculated;
    value             $internalField;
```

```
43    }
44  }
```

A.2.3 0/p_rgh

```
1   FoamFile
2   {
3       version     2.0;
4       format      ascii;
5       class       volScalarField;
6       object      p_rgh;
7   }
8   dimensions          [1 -1 -2 0 0 0 0];
9
10  internalField       uniform 1e5;
11
12  boundaryField
13  {
14    inlet
15    {
16      type              fixedFluxPressure;
17      value             $internalField;
18    }
19
20    inlet_water
21    {
22      type              fixedFluxPressure;
23      value             $internalField;
24    }
25
26    outlet
27    {
28      type              fixedValue;
29      value             uniform 1e5;
30    }
31
32    walls
33    {
34      type              zeroGradient;
35    }
36
37    stream
38    {
39      type              zeroGradient;
40    }
41  }
```

A.2.4 0/U.air

```
1   FoamFile
2   {
3       version     2.0;
4       format      binary;
5       class       volVectorField;
6       object      U.air;
7   }
8
9   dimensions      [0 1 -1 0 0 0 0];
10
11  internalField   uniform (2 0 0);
12
13  boundaryField
14  {
15    inlet
16    {
17      type              fixedValue;
18      value             uniform (2 0 0);
19    }
20
21    inlet_water
22    {
23      type              fixedValue;
24      value             uniform (0 0 0);
25    }
26
27    outlet
28    {
29      type                inletOutlet;
30      value             $internalField;
31      inletValue        uniform (0 0 0);
32    }
33
34    walls
35    {
36      type              slip;
37      value             uniform (0 0 0);
38
39    stream
40    {
41      type              zeroGradient;
42    }
43  }
```

A.3 Specific files for DPMFoam

A.3.1 0/p

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       volScalarField;
    object      p;
}

dimensions      [0 2 -2 0 0 0 0];

internalField   uniform 1e5;

boundaryField
{
    inlet
    {
      type
          fixedFluxPressure;
      value              $internalField;
    }

    inlet_water
    {
      type
          fixedFluxPressure;
      value              $internalField;
    }

    outlet
    {
          type            fixedValue;
          value           1e5;
    }

    walls
    {
      type               zeroGradient;
    }

    stream
    {
      type               zeroGradient;
    }
}
```

B

twoPhaseEulerFoam configuration files

B.1 constant/phaseProperties

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      phaseProperties;
}
phases (air water);

air
{

  /*Comment to enable IAC equation */
  diameterModel       constant;
  constantCoeffs
  {
    d                 1e-3;
  }

  residualAlpha       1e-6;

  /* Uncomment to enable IAC equation */

  // diameterModel   IATE;
  // IATECoeffs
  // {
  //   dMax              1e-1;
  //   dMin              1e-5;
  //   residualAlpha     1e-6;
  //
  //   sources
  //   (
  //     wakeEntrainmentCoalescence
  //     {
  //       Cwe           0.0073;
  //     }
  //
  //     randomCoalescence
  //     {
  //       Crc           0.021;
  //       C             3;
  //       alphaMax      0.8;
  //     }
  //
  //     turbulentBreakUp
  //     {
  //       Cti           0.0945;
  //       WeCr          2;
  //     }
  //   );
  // }
  //
  // residualAlpha       1e-6;
}

/* Just here for compatibility, not used
    */
water
{
  diameterModel       constant;
  constantCoeffs
  {
    d                 1e-4;
  }

  residualAlpha       1e-6;
}

blending
{
  default
  {
    type              none;
    continuousPhase   water;
  }
}

sigma
(
  (air and water)     0.07
);
```

```
82
83  aspectRatio
84  (
85    (air in water)
86    {
87      type            constant;
88      E0              1.0;
89    }
90  );
91
92  drag
93  (
94    (air in water)
95    {
96      type            SchillerNaumann;
97      residualRe      1e-3;
98      swarmCorrection
99      {
100       type          none;
101     }
102   }
103 );
104
105 virtualMass
106 (
107   (air in water)
108   {
109     type
            constantCoefficient;
110     Cvm             0.5;
111   }
112 );
113
114 heatTransfer
115 (
116   (air in water)
117   {
118     type            RanzMarshall;
119     residualAlpha   1e-4;
120   }
121 );
122
123 lift
124 (
125   (air in water)
126   {
127     type
            constantCoefficient;
128     Cl              0.5;
129   }
130 );
131
132 wallLubrication
133 (
134   (air in water)
135   {
136     type            Frank;
137     Cwd             6.8;
138     Cwc             10;
139     p               1.7;
140   }
141 );
142
143 turbulentDispersion
144 (
145   (air in water)
146   {
147     type            LopezDeBertodano;
148     Ctd             1;
149   }
150 );
151
152 // Minimum allowable pressure
153 pMin             10000;
```

B.2 constant/thermophysicalProperties.air

```
1   FoamFile
2   {
3       version     2.0;
4       format      ascii;
5       class       dictionary;
6       location    "constant";
7       object      thermophysicalProperties
            .air;
8   }
9
10  thermoType
11  {
12      type            heRhoThermo;
13      mixture         pureMixture;
14      transport       const;
15      thermo          hConst;
16      equationOfState rhoConst;
17      specie          specie;
18      energy
            sensibleInternalEnergy;
19  }
20
21  mixture
22  {
23      specie
24      {
25          nMoles      1;
26          molWeight   28.9;
27      }
28          equationOfState
29      {
30         rho 1.269;
31      }
32      thermodynamics
```

```
    {
        Cp          1007;
        Hf          0;
    }
    transport
    {
        mu          1.725e-05;
        Pr          0.7;
    }
}
```

B.3 constant/thermophysicalProperties.water

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      thermophysicalProperties
        .water;
}

thermoType
{
    type            heRhoThermo;
    mixture         pureMixture;
    transport       const;
    thermo          hConst;
    equationOfState rhoConst ;
    specie          specie;
    energy
        sensibleInternalEnergy;
}

mixture
{
    specie
    {
        nMoles      1;
        molWeight   18;
    }
    equationOfState
    {
       rho 1000;
    }
    thermodynamics
    {
        Cp          4195;
        Hf          0;
    }
    transport
    {
        mu          1e-3;
        Pr          2.289;
    }
}
```

B.4 constant/turbulenceProperties.water

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties.
        water;
}

simulationType  RAS;

RAS
{
    RASModel  kEpsilon; //LaheyKEpsilon;

    turbulence      on;
    printCoeffs     on;
}
```

B.5 constant/turbulenceProperties.air

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
```

```
    object      turbulenceProperties.air;
}

simulationType  RAS;

RAS
{
    RASModel continuousGasKEpsilon; // continuousGasKEpsilon;

    turbulence      off;
    printCoeffs     on;
}
```

B.6 system/controlDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}

application     twoPhaseEulerFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         5;

deltaT          0.0005;

writeControl    runTime;

writeInterval   0.1;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

adjustTimeStep  yes;

maxCo           0.5;
maxDeltaT       0.2;
```

B.7 system/fvSchemes

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSchemes;
}
ddtSchemes
{
  default             Euler;
}

gradSchemes
{
  default             Gauss linear;
}

wallDist
{
  method            meshWave;
  nRequired         false;
}

divSchemes
{
  default             none;

  div(phi,alpha.air)  Gauss vanLeer;
  div(phir,alpha.air) Gauss vanLeer;

  "div\(alphaRhoPhi.*,U.*\)"      Gauss limitedLinearV 1;
  "div\(phi.*,U.*\)"              Gauss limitedLinearV 1;
   div(phi.air,kappai.air)        Gauss vanLeer;
  "div\(alphaRhoPhi.*,(h|e).*\)"  Gauss
```

```
        limitedLinear 1;
    "div\(alphaRhoPhi.*,K.*\)"      Gauss
        limitedLinear 1;
    "div\(alphaPhi.*,p\)"           Gauss
        limitedLinear 1;
    "div\(alphaRhoPhi.*,(k|epsilon).*\)"
        Gauss limitedLinear 1;
    "div\(phim,(k|epsilon)m\)"      Gauss
        limitedLinear 1;
    "div\(\(\(\(\(alpha.*\*thermo:rho.*\)*
        nuEff.*\)\*dev2\(T\(grad\(U.*\)\)
        \)\)\)\)" Gauss linear;
}

laplacianSchemes
{
    default             Gauss linear
        uncorrected;
}

interpolationSchemes
{
    default             linear;
}

snGradSchemes
{
    default             uncorrected;
}
```

B.8 system/fvSolution

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}

solvers
{
    alpha.air
    {
        nAlphaCorr      1;
        nAlphaSubCycles 2;
        MULESCorr       yes;
        nLimiterIter    8;
    }

    p_rgh
    {
        solver          GAMG;
        smoother        DIC;
        nPreSweeps      0;
        nPostSweeps     2;
        nFinestSweeps   2;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
        tolerance       1e-8;
        relTol          0;
    }

    p_rghFinal
    {
        $p_rgh;
        relTol          0;
    }

    "(U|kappai).*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-5;
        relTol          0;
        minIter         1;
    }
    "(k|epsilon|Theta).*"
 {
     solver          smoothSolver;
     smoother        symGaussSeidel;
     tolerance       1e-7;
     relTol          0;
     minIter         1;
 }
}

PIMPLE
{
  nOuterCorrectors 5;
  nCorrectors      2;
  nNonOrthogonalCorrectors 0;
  residualControl
  {
  p_rgh
        {
        tolerance  1e-3;
        relTol      0;
        }
  }
}

relaxationFactors
{
    equations
    {
        "U.*"           0.4;
       "kappai.*"            0.4;
    }
}
```

C

DPMFoam configuration files

C.1 constant/kinematicCloudProperties

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      particleProperties;
}


solution
{
    active          true;
    coupled         true;
    transient       yes;
    cellValueSourceCorrection off;

    interpolationSchemes
    {
        rho.water         cell;
        U.water           cellPoint;
        mu.water          cell;
        curlUcDt cell;
        DUcDt cell;
    }

    integrationSchemes
    {
        U               Euler;
    }

    sourceTerms
    {
        schemes
        {
            U semiImplicit 1;
        }
    }
}

constantProperties
{
    parcelTypeId 1;

    rhoMin          1e-15;
    minParcelMass   1e-15;

    rho0            1.2;
    youngsModulus   1e-3;
    poissonsRatio   0.5;

    constantVolume  false;

    alphaMax        0.99;
}

subModels
{
    particleForces
    {
          sphereDrag
        {
            alphac alpha.water;
        }
        gravity;
       ConstantLift{ U U.water;}
       virtualMass{ Cvm 0.5; U U.water;}
    }

    injectionModels
    {
        model1
        {
            type
                patchInjection;
            parcelBasisType fixed;
            patchName       inlet;
            U0              (2 0 0);
            nParticle       1;
            parcelsPerSecond 2521;
            sizeDistribution
            {
                type       fixedValue;
                fixedValueDistribution
                {
                  value 1e-3;
                }
            }
```

```
            flowRateProfile constant 0.00132;
            massTotal       0;
            SOI 0;
            duration 60;
        }
    }

    dispersionModel gradientDispersionRAS;

    patchInteractionModel localInteraction;

    localInteractionCoeffs
    {
        patches
        (
            walls
            {
                type rebound;
                e    0.97;
                mu   0.09;
            }

            "inlet|inlet_water|outlet|stream"
            {
                type escape;
            }
        );
    }

    StandardWallInteractionCoeffs
    {
        type rebound;
        e    0.97;
        mu   0.09;
    }

    heatTransferModel none;

    surfaceFilmModel none;

    collisionModel pairCollision;

    pairCollisionCoeffs
    {
        maxInteractionDistance  5e-4;

        writeReferredParticleCloud no;

        pairModel pairSpringSliderDashpot;

        pairSpringSliderDashpotCoeffs
        {
            useEquivalentSize   no;
            alpha               1;
            b                   1;
            mu                  0;
            cohesionEnergyDensity 0;
            collisionResolutionSteps 12;
        };

        wallModel wallSpringSliderDashpot;

        wallSpringSliderDashpotCoeffs
        {
            useEquivalentSize no;
            collisionResolutionSteps 12;
            youngsModulus   1e-3;
            poissonsRatio   0.50;
            alpha          1;
            b               1;
            mu              0;
            cohesionEnergyDensity 0;
        };

        UName U.water;
    }

    stochasticCollisionModel none;

    radiation off;
}


cloudFunctions
{}
```

C.2 constant/transportProperties

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      transportProperties;
}

continuousPhaseName water;

rho.water        1000;

transportModel  Newtonian;
nu              1e-06;
```

C.3 constant/turbulenceProperties.water

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "constant";
    object      turbulenceProperties.water;
}

simulationType  RAS;

RAS
{
    RASModel  kEpsilon;

    turbulence      on;
    printCoeffs     on;
}
```

C.4 system/controlDict

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      controlDict;
}

application     DPMFoam;

startFrom       startTime;

startTime       0;

stopAt          endTime;

endTime         8;

deltaT          2e-3;

writeControl    runTime;

writeInterval   0.05;

purgeWrite      0;

writeFormat     ascii;

writePrecision  6;

writeCompression uncompressed;

timeFormat      general;

timePrecision   6;

runTimeModifiable yes;

adjustTimeStep  no;

maxCo           0.9;
maxDeltaT       0.2;
```

C.5 system/fvSchemes

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    object      fvSchemes;
}

ddtSchemes
{
    default Euler;
}

gradSchemes
{
    default         Gauss linear;
}

divSchemes
{
    default         none;

    div(alphaPhic,U.water)    Gauss
        linearUpwindV unlimited;
    div(alphaPhic,epsilon.water) Gauss
        limitedLinear 1;
```

```
    div(alphaPhic,k.water) Gauss limitedLinear 1;
    div(((alpha.water*nuEff.water)*dev2(T(grad(U.water))))) Gauss linear;
}

laplacianSchemes
{
    default         Gauss linear corrected;
}

interpolationSchemes
{
    default         linear;
}

snGradSchemes
{
    default         corrected;
}
```

C.6 system/fvSolution

```
FoamFile
{
    version     2.0;
    format      ascii;
    class       dictionary;
    location    "system";
    object      fvSolution;
}

solvers
{
    p
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0.01;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    pFinal
    {
        solver          GAMG;
        tolerance       1e-06;
        relTol          0;
        smoother        GaussSeidel;
        cacheAgglomeration true;
        nCellsInCoarsestLevel 10;
        agglomerator    faceAreaPair;
        mergeLevels     1;
    }

    "(U|k|epsilon).*"
    {
        solver          smoothSolver;
        smoother        symGaussSeidel;
        tolerance       1e-05;
        relTol          0.1;
    }

    "(U|k|epsilon).*Final"
    {
        $U;
        tolerance       1e-05;
        relTol          0;
    }
}

PIMPLE
{
  pRefPoint (0 0 0);
  pRefValue 0;
  nOuterCorrectors 5;
  nCorrectors      2;
  nNonOrthogonalCorrectors 0;
  residualControl
  {
  p
{
        tolerance  1e-3;
        relTol      0;
}
}
}

relaxationFactors
{
    equations
    {
        "U.*"           0.4;
       "kappai.*"            0.4;


    }
}
```