NTNU – Trondheim
Norwegian University of
Science and Technology

Numerical study of the droplet-interface dynamic related to liquid-liquid separators

Emile Simen Augustin
Rusten

Master of Science in Physics and Mathematics
Submission date: June 2013
Supervisor: Jon Andreas Støvneng, IFY
Co-supervisor: Carlos Dorao, EPT

Norwegian University of Science and Technology
Department of Physics

Preface

This master's thesis is the concluding work for a Master of Science and Technology in Technical Physics from the Norwegian University of Science and Technology in Trondheim. During the Spring 2013 and under the guidance of Pr. Dorao, I have worked on numerical simulations of water drops in viscous oils. The thesis is submitted at the Department of Physics of NTNU but was done on behalf of the Natural Gas Technology research group at the Department of Energy and Process Engineering at NTNU. The workload is twenty weeks, considered to be equivalent to one semester of full-term studies.

I would like to thank my supervisor Pr. Dorao for useful guidance throughout the semester and Pr. Støvneng for helping me in the difficult exercise of presenting scientific results in a clear manner. I would also like to thank all lecturers and employees of NTNU for making my five years of studies interesting and fun.

Finally, I would like to thank my family for continuous support and all of those who helped me have a social life besides this Master's thesis. I had great times with Carl-Philip and his roommates and without sharing ice creams, pancakes, smoothies and cakes with them, this year would not have been as enjoyable as it was. I would especially like to thank my supercalifragilisticexpialidocious girlfriend Nadir, not for the frequent tickling assaults she inflicts me, but for all the good times and fits of laughter we had together.

"Happiness is only real when shared."

– Alexander Supertramp

Emile Rusten
June 16, 2013.

Abstract

The terminal velocity of liquid drops in a continuous liquid phase occurs in several industrial applications. Specifically, the separation of oil and water when extracting crude oil from offshore or land reservoirs, is crucially dependent on this parameter. Little information in the literature and the lack of experimental studies motivate the research in this field. Instead of setting up an experiment, one can run a numerical simulation to find results. This saves time and money and is sometimes the only solution for cases which are impossible to test experimentally.

We have used the CFD package OpenFoam with a VOF approach to simulate the fall of water drops in Exxsol D80 oil, with diameters ranging from 100 µm to 1000 µm. A literature review showed the presence of parasitic currents in VOF multiphase simulations. Their magnitude is therefore studied to evaluate how much such currents affect the terminal velocity. The study of the magnitude of parasitic current was done by varying the viscosity and the surface tension of the drop.

Our results showed good agreement with analysis of parasitic currents in the literature. Parasitic currents magnitude in OpenFoam ranged from 10% to 100% or more of the simulated terminal velocity. Low surface tension and high drop viscosity reduced parasitic current to 10% of the simulated terminal velocity. When using the physical parameters of water and Exxsol D80 oil, results were of poor quality: parasitic current magnitude was over 50% of the simulated terminal velocity for diameters equal to 500 µm and 1000 µm. Increasing the drop viscosity to hundred times the viscosity of water showed good agreement with theoretical predictions on the terminal velocity of solid spheres.

This work shows that OpenFoam can be potentially generalized to simulate the fall of liquid drops in a liquid phase. One should be careful when using surface tension dominated flows. Results can be exploitable, but preliminary validation is essential. For the water-D80 system, parasitic currents were too strong to yield exploitable results. However, decreasing surface tension and increasing the drop viscosity, resulted in simulating a case close to a rigid sphere. This diminished parasitic currents and showed good agreement with theoretical and experimental results.

Sammendrag

Terminalhastigheten til dråper av en væske i en annen er veldig viktig i mange industrielle prosesser. Denne parameteren er kritisk i dimensjonering og ytelsesforbedring av væske-væskeseparatorer i olje-og gassindustrien. Det er lite informasjon i litteraturen og for få eksperimentelle forsøk har blitt utført for vanndråper i olje og for rekkevidden til dråpediameteren som finnes i væske-væskeseparatorer. Ved å bruke numeriske simulasjoner på slike systemer kan man unngå å sette opp et eksperimentelt forsøk. Dette kan spare tid og penger, i tillegg til at numeriske beregninger kan simulere systemer som er umulig å teste eksperimentelt.

Vi vurderer bruken av det numeriske verktøyet OpenFoam med en VOF-metode for å simulere synkingen av vanndråper i Exxsol D80 olje. Rekkevidden på diameteren som skal undersøkes er fra $d = 100$ µm til $d = 1000$ µm. Et litteraturstudie viser at parasitiske strømninger påvirker flerfasesimuleringer med VOF-metoden. Oppgaven vil også studere betydningen av slike strømninger for å kunne vurdere resultatenes kvalitet. Dette blir gjort ved å endre dråpeviskositeten og overflatespenningen.

Numeriske resultater for parasitiske strømninger stemmer med analyse gjort av andre forfattere. Dråper med lav overflatespenning eller høy viskositet skaper parasitiske strømninger under 10% av slutt-hastigheten. For vanndråper i Exxsol D80 olje og med diameteren brukt for denne oppgaven, kunne ikke resultatene bli utnyttet på grunn av høy parasitiske strømninger. Parasitiske strømninger ble målt til over 50% av dråpens terminalhastighet. Ved å øke dråpeviskositeten til hundre ganger viskositeten til vann, fulgte resultatene teoretiske lover for faste dråper.

OpenFoam med en VOF-metode kan bli brukt til å simulere dråper i en annen væske. Brukeren må være forsiktig med strømninger med høy overflatespenning og innledende validering er uansett nødvendig. For vanndråper i Exxsol D80 olje og med diameteren brukt for denne oppgaven, kunne ikke resultatene utnyttes på grunn av høye parasitiske strømninger. Ved å bruke lavere overflatespenning og høyere dråpeviskositet ble resultatene gode og stemte overens med teoretiske lover for faste sfæriske partikler.

Contents

List of Figures

List of Tables

List of Codes

List of Symbols

Normal symbols represent scalar quantities and boldface symbols represent vector and tensor quantities.

Variables

Symbol	Description	Units
ρ	Density	$[\mathrm{kg\,m^{-3}}]$
p	Pressure	[Pa]
g	Acceleration of gravity	$[\mathrm{m\,s^{-2}}]$
$\mathbf{v}$, u, U	Velocity field, velocity magnitude	$[\mathrm{m\,s^{-1}}]$
Ω	Domain	[−]
$\mathbf{T}$	Viscous stress tensor	[Pa s]
ν	Kinematic viscosity	$[\mathrm{m^2\,s^{-1}}]$
μ	Dynamic viscosity	[Pa s]
σ	Interfacial tension	$[\mathrm{N\,m^{-1}}]$
α	Phase fraction	[−]
d, d_d, D	Drop diameter	[m]
R, r	Drop radius	[m]
C_D	Drag coefficient	[−]
Re	Reynolds number	[−]
Eo	Eötvös number	[−]
Mo	Morton number	[−]
We	Weber number	[−]
Co	Courant number	[−]
Δx, Δy	Horizontal, vertical grid spacing	[m]
Δt, t	Time spacing, time step	[s]
A	Area	$[\mathrm{m^2}]$
a, b	Minor, major axis	[m]
κ	Curvature	$[\mathrm{m^{-1}}]$
$\mathbf{n}$	Normal vector	[−]
Subscript	**Description**	
d	Drop	
s	Sphere	

Continued on next page

Continued from previous page

t	Terminal	
w	Water	
D80	Exxsol D80 oil	
po	Paraffin oil	
c	Continuous phase	
d	Dispersed phase	
p, para	Parasitic	
max	Maximum	
Acronyms	**Description**	
CFD	Computational Fluid Dynamics	
LS	Level set	
VOF	Volume of Fluid	
NSE	Navier-Stokes Equations	
2D	Two-dimensional	
CSF	Continuum Surface Force	
NTNU	Norwegian University of Science and Technology	
CFL	Courant-Friedrichs-Lewy	

CHAPTER 1

Introduction

The terminal velocity of drops in an ambient fluid is of great interest. They occur in numerous industrial processes such as the food, pharmaceutical, chemical, and petroleum industry. Liquid-liquid separators and three-phase separators in the oil and gas industry, which are the background for this work, are very much dependent on this key parameter.

Numerical simulations are nowadays commonly used in fluid mechanics because computing power has become easily available in the past decade. If results from simulations are of good quality and are in agreement with experimental data, one can save time and money by running simulations instead of setting up an experimental study. Even if one should always be critical about Computational Fluid Dynamics (CFD) simulations, results can give explanations to different phenomena or at least give some indications. Furthermore, there is always limitations in experimental setups, while numerical simulations can be easily standardized or used to study complex cases which can not be studied experimentally.

1.1 Background

1.1.1 Liquid-liquid separators

The oil coming out from a well is never pure, but usually a mixture of gas, water, oil, and solid particles. This mixture needs to be separated for further processing. Stewart and Arnold [34] name some of the motivations behind separating phases:

- Equipment downstream of the well can only operate with one phase. Pumps for example, can not work efficiently in the presence of gas, while measuring devices are inaccurate when water is dispersed in the oil phase. Compressor and dehydration systems require liquid-free gases.
- The final product needs to meet quality, safety, and environmental specifications on impurities. Usually, it is required that the oil contains less than 0.5% water [28].

Physical separation of the three immiscible phases takes place in separators and usually relies on one or more of three mechanisms: momentum shift, coalescence,

and gravity settling. The mechanism of a three-phase horizontal separator will be briefly presented to justify the study of terminal velocity of drops in an ambient fluid.

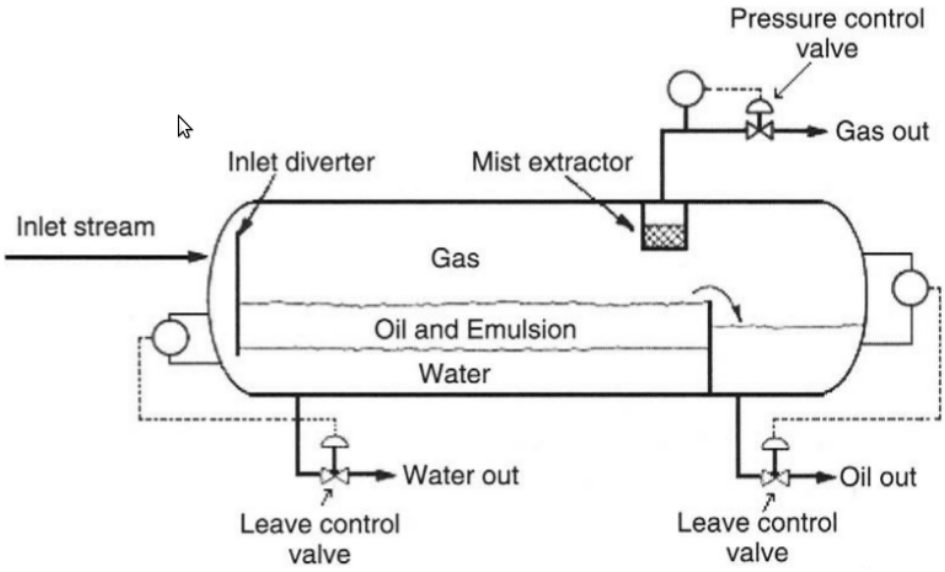


Figure 1.1: Side view of a horizontal three-phase separator. The gas, oil, and water mixture enters in the inlet stream. Figure is from *Handbook of Natural Gas Transmission and Processing* by Mokhatab et al. [27].

Mokhatab et al., in the *Handbook of Natural gas transmission and processing* [27], explain how separators work, and their explanations will be paraphrased. Looking at the side view of a three-phase separator in Figure 1.1, one can see the inlet diverter which will change the direction of the entering flow. Momentum change can be used to remove most of the gas from the liquid phase. Because the inertia of liquid particles is greater than the gas, liquid will tend to flow in the original direction whereas the gas will be directed away easily.

The inlet flow will often be directed through the water phase in the bottom of the vessel for "water washing". The objective is to force the inlet mixture to mix with water, thus promoting coalescence of water droplets. Coalescence is the merging of a drop with the same phase[1]. Another reason is to avoid the presence of water in the top layer: it contains already "clean" oil that is ready to be pumped out and it is undesirable to contaminate this phase.

The next step is to let gravity act on the density difference between oil and water: water drops in the emulsion phase will sink to the water phase. Gas is evacuated through the mist extractor, while the water level can control which part

[1] which can be another drop or a liquid film

of the oil layer can be sent out of the vessel. Residence time can vary from minutes to tens of minutes [27].

The mechanisms presented here are simplified, as several techniques are used to improve the dimensions of vessels and the time needed for separation. Using baffles to avoid turbulence and facilitate coalescence can be mentioned [10].

1.1.2 Motivation of the work

During the last phase of the oil and water separation, the diameter of the water drop is of considerable importance. The general trend found by experimental studies in the past years shows that the terminal velocity is increased when the diameter gets bigger. There is of course an interest in understanding the dynamics of drop falls, as it may lead to optimized separators. Settling time is related to the vessel size which is often a problem on subsea installations or offshore platforms where space is scarce. An optimized separator, with reduced size or more efficient in separating fluids is therefore important for practical and economical reasons.

The study of terminal velocity of drops is motivated by the lack of information in the literature. Additionally, too few experiments have been performed on falling drops of water in viscous oils, with a drop diameter within the range observed in liquid-liquid separators. To our knowledge, the fall of drops with diameter ranging from 100 µm to 1000 µm and with high surface tension has never been simulated. Furthermore, a qualitative or quantitative analysis of parasitic currents in OpenFoam, essential to evaluate the quality of the results, has not been found in the literature.

1.2 Objective of the work

The main objective of the work is to investigate the possibility of using Computational Fluid Dynamics (CFD) to simulate the fall of small water drops in oil.

The main challenge of numerical simulations is to validate the results. Specifically, the small size of the drop and the high surface tension, will create parasitic currents. Such currents need to be studied to evaluate their effect. Results also need to be confronted with experimental data or results from published scientific articles and theoretical predictions.

1.3 Scope of the work

The CFD package OpenFoam will be used to simulate the fall of liquid drops in an ambient fluid. This work is based on an experimental setup at NTNU, where water drops fall in Exxsol D80 oil. Drop diameter in the experimental setup at NTNU ranges from 100 µm to 1000 µm and this will also be the range considered in the simulations. The viscosity of the ambient fluid will always be the viscosity

of Exxsol D80 oil, but the viscosity of the drop and the surface tension will be varied to study parasitic currents. To validate the simulations, some results will be confronted to existing results in the literature.

1.4 Report structure

The report will be divided in three parts. The first part will be theoretical. Chapter 2 presents existing results from the literature which are relevant to the domain of terminal velocity of liquid drop in an ambient liquid. Then, some theory about the model and equations linked to the project will be explained in Chapter 3. The next part, Chapter 4 will present the settings of the simulation. Finally, Chapter 5 will present and comment results of the simulations and our conclusions are collected in Chapter 6. Appendix A will show some important codes that were frequently used or specific to OpenFoam to obtain the results.

CHAPTER 2

Literature review

The terminal velocity of drops in an ambient fluid has been extensively studied in the past decades by the scientific community. Before computing power was available for simulations, several experimental studies were performed. This chapter will review the existing literature in this domain, both for experimental and numerical studies, and some important results will be mentioned. At the end of this chapter, Table 2.1 shows the main literature used for this work.

2.1 Experimental studies on terminal velocity

The terminal velocities of liquid drops in a liquid ambient phase have been studied experimentally in the past. Stationary conditions were usually considered for these studies. Hu and Kintner [19] were some of the first to publish correlations for terminal velocity. They studied liquids with high surface tension. On the other hand, Klee and Treybal [20] derived correlations applicable for systems with medium surface tension. The behaviour of rising droplets with oscillations and recirculation was taken into account. One can also mention Licht and Narasimhamurty [26] who studied organic liquids falling through water. These authors found that terminal velocity is increased for larger drops. Also, smaller drops tend to keep a spherical shape through the fall. Several other studies were performed, with different Reynolds, Eötvös, and Morton numbers. Changing these dimensionless numbers is equivalent to varying the densities, viscosities, and interfacial tension of the two phases. Figure 2.1 shows the different regimes of bubble shape.

Experimental studies on solid spheres have also been performed in the past. Theoretical laws on terminal velocity of solid particles, presented in Chapter 3, are based on such experimental data.

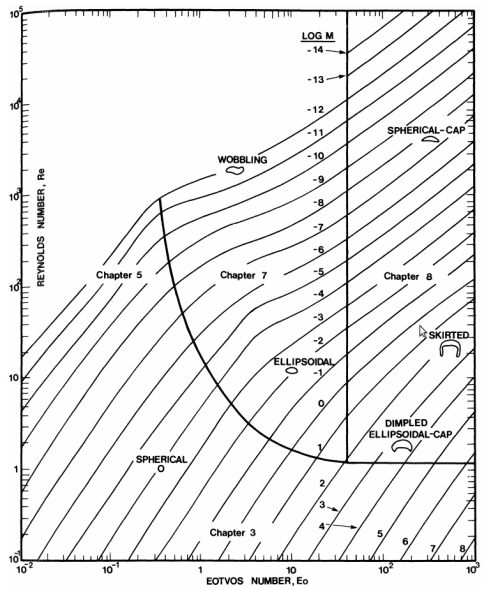


Figure 2.1: Shape regimes for bubbles and drops in unhindered gravitational motion through viscous liquids. Graph is from *Bubbles, drops, and particles* by Clift et al. [9].

2.2 Numerical experiments on terminal velocity

The increase in computational power in the last decades has allowed numerical simulations to be an important part of research in multiphase flows. CFD methods allow the user to simulate moving interfaces, such as the motion of a drop in a continuous phase. Different numerical methods and solving strategies exist which are able to capture and describe phenomena at and around the interface.

There are two general strategies to simulate interfaces: *front capturing methods* and *front tracking methods*. Other methods combine features from both, but are not within the scope of this project.

The following sections will present the two main approach for simulating interfaces. A literature review of existing work will be done, mainly focused on Volume of Fluid (VOF) simulations because it is the approach chosen for this work. Authors using other CFD packages will also be mentioned.

2.2.1 Front tracking methods

In front tracking methods, a moving mesh describes the motion of the interface. As the topology of one phase changes, the mesh is moved along with the interface. These methods require each their sets of equations which describe the flow field. These methods are usually very robust, but are on the other hand rather difficult to implement in a code [14].

2.2.2 Front capturing methods

In front capturing methods, a fixed mesh[1] is used, while special functions or markers are used to track the interface on the mesh. One of the most popular surface capturing methods is the Level Set (LS) method presented by Osher and Fedkiw [30]. This method uses an indicator function which takes positive and negative values on each side of the interface and is zero on the interface. This method is easy to implement but its main drawback is the loss of mass for interfaces that are very irregular [14].

Another popular method is the Volume of Fluid (VOF) method introduced by Hirt and Nichols [18]. Such methods use a volume fraction function. For a given cell, this function takes a value of zero for the ambient continuous phase, and a value of one for the dispersed phase (i.e. the drop). Cells close to the interface have values between zero and one. The main drawback of this method is a numerical smearing of the interface [14]. This is the method used for the present work, and will be presented in detail in Chapter 3, Section 3.2.

2.2.3 Existing numerical studies on falling or rising drops

Deshpande and Zimmerman [13] studied the motion of interface for buoyancy-driven droplets using a Level-Set method. Results were influenced by the Reynolds number (Re), the Eötvös number (Eo), and the Morton number (Mo). They considered different sets of these dimensionless numbers. Results regarding droplet shape were in good agreement with experimental results of Clift et al. [9].

Bäumler et al. [6] used the academic code NAVIER with VOF to simulate organic droplets rising in water. Diameter ranged from $d = 1$ mm to $d = 10$ mm. The phases considered in the article are toluene/water, n-butyl acetate/water, and n-butanol/water. The interfacial tension ranged from $\sigma = 0.00163\ \mathrm{N\,m^{-1}}$ to $\sigma = 0.033\ \mathrm{N\,m^{-1}}$. Numerical results on droplet shape and terminal velocities were in good agreement with several results from the literature, as well as from their own experimental setup.

Similarly, Eiswirth et al. [14] simulated the rise of toluene in an aqueous phase using the VOF approach implemented in the CFD tool COMSOL Multiphysics 3.3a. The range of droplet diameter used is similar to the article of Bäumler et al. They also performed an experimental study which showed excellent agreement in terms of the droplet shape.

Finally, several authors published results which are close to this work; the VOF approach in OpenFoam were used by the following authors: Klostermann et al. [21] simulated rising bubbles with an ellipsoid and skirted shape, and compared the results to benchmark tests done by other research groups. Results were in reasonable agreement with previous results, but the authors recommend users to be careful for surface tension dominated flows.

[1] also called Eulerian mesh

Samkhaniani et al. [32] used OpenFoam to study the oscillations of bubbles. The shape of the droplet was also checked for radii ranging from $d = 2$ mm to $d = 62$ mm. The Eötvös and Morton numbers were varied to test different regimes, ellipsoidal, spherical cap, and skirted regime for example. Their results showed good agreement quantitatively and qualitatively in most bubble regimes, but for high Reynolds number the author reports the trajectory of the bubble was no longer straight.

Lekhilifi et al. [25] used a VOF approach to simulate a droplet of water with diameter $d = 2$ mm falling in paraffin oil. The solver used is not mentioned. The authors used a surface tension $\sigma = 0.02\ \mathrm{N\,m^{-1}}$ and studied the terminal velocity, wall effects, and internal circulation in a single drop. The author compare results to the Hadamard-Rybczynski law, but not to experimental results.

2.3 Numerical studies of parasitic currents

The implementation of surface tension model in a VOF method will generate non-physical parasitic currents which will be presented in detail in Chapter 3, Section 3.2.

Evidence of parasitic currents in VOF simulations was first discovered in 1994 by Lafaurie et al. [24] who coined the term parasitic currents[2]. Scardovelli and Zaleski [33] in 1999 also studied the phenomenon briefly. In 2007, Harvie et al. [16] did a quantitative analysis of the magnitude of these parasitic currents.

2.4 Concluding remarks

The fall of liquid drops has been extensively studied in the past, both experimentally and numerically. However, to our knowledge, drops with diameter ranging from 100 µm to 1000 µm and with high surface tension have never been simulated. Furthermore, no analysis of parasitic currents in OpenFoam has been found in the literature. The literature review supports that the project is useful and unique. It is interesting to simulate drops in the range considered and study the effect of parasitic currents using OpenFoam.

Table 2.1: Overview of the literature related to the present work

Author	**Title of work**	**Key words**
Stewart and Arnold, 2008	Gas-Liquid and Liquid-Liquid Separators	Description of separators
Mokhatab et al., 2006	Handbook of Natural Gas transmission and processing	Role of components in a separator

Continued on next page

[2] also called spurious currents

Table 2.1 – *Continued from previous page*

Author	**Title of work**	**Key words**
Cusak, 2009	Rethink your liquid-liquid separation	Improvement on settling time with coalescers
Hu and Kintner, 1955	The fall of single liquid drops through water	Experimental study on terminal velocity, high surface tension
Klee and Treybal, 1956	Rate of rise or fall of liquid drops	Experimental study on terminal velocity, medium surface tension
Licht and Narasimhamurty, 1955	Rate of fall of single liquid droplets	Experimental study on terminal velocity
Kumar, 1983	Droplet behaviour in liquid/liquid extraction	Drag force, recirculation, terminal velocity
Eiswirth et al., 2011	Experimental and numerical investigation of a free rising droplet	COMSOL Multiphysics, Level set approach, toluene/water
Osher and Fedkiw, 2001	Level set methods: an overview and some recent results	Level set method
Hirt and Nichols, 1979	Volume of Fluid (VOF) method for the dynamics of free boundaries	Numerics of VOF approach
Lekhlifi et al., 2009	Numerical simulation of the unsteady hydrodynamics of a water droplet in paraffin oil	Terminal velocity, VOF
Klostermann et al., 2012	Numerical simulation of a single rising bubble by VOF with surface compression	VOF, OpenFOAM, CSF
Samkhaniani et al., 2012	Direct Numerical Simulation of Single Bubble Rising in Viscous Stagnant Liquid	VOF, drop oscillations , OpenFOAM
Perry, 2007	Chemical Engineers Handbook	Terminal velocity, solid sphere, Intermediate law, Stokes' law
Clift et al., 2005	Bubbles, Drops and Particles	Terminal velocity, Hadamard-Rybczynski equation, drop shape
Brackbill et al., 1991	A continuum method for modelling surface tension	CSF approach
Lafaurie et al., 1993	Modelling merging and fragmentation in multiphase flow with SURFER	CSF, parasitic currents, VOF

Continued on next page

Table 2.1 – *Continued from previous page*

Author	**Title of work**	**Key words**
Scardovelli and Zaleski, 1999	Direct numerical simulation of free-surface and interfacial flow	CSF, parasitic (spurious) current
Harvie et al., 2006	An analysis of parasitic current generation in Volume of Fluids simulations	Magnitude of parasitic currents for different regimes
Marquez Damian, 2009	Description and utilization of interFoam multiphase solver	VOF, phase fraction, CSF, numerical schemes
Versteeg and Malalasekera, 2007	An Introduction to Computational Fluid Dynamics	NSE, discretization, flux, numerical schemes
White, 2003	Fluid Mechanics	Momentum equation, continuity equation

CHAPTER 3

Theory

This chapter will present the theory behind this thesis. The Navier-Stokes equations describe the flow of fluids and are the core of any problem in fluid mechanics. These equations describe the fall of a drop in a fluid and are included in the OpenFoam code: the equations will therefore be presented. Some theory about the numerical aspect of solving the Navier-Stokes equations using the Volume of Fluid (VOF) approach will also be presented. The modelling of surface tension in OpenFoam will be explained, and because this leads to parasitic currents, some explanations of their origin will also be included in this chapter. Finally, some theoretical laws on the terminal velocity of solid or liquid drops in an ambient fluid will be presented.

3.1 The Navier-Stokes equations

This section will introduce the main equations of this work: the celebrated Navier-Stokes equations (NSE) describe the flow of the problem and are the backbone of all CFD codes. These equations are discretized in space and time and solved numerically, to describe the evolution of the flow in the domain of interest. It is assumed the fluids are immiscible and that they have constant viscosity and density[1]. Other assumptions are that the temperature is constant and no chemical reactions are occurring between the two fluids.

Assuming the fluids are incompressible, Newtonian, and homogeneous, the NSE equations read [39]:

$$\nabla \cdot \mathbf{v} = 0 \tag{3.1}$$

$$\rho(\frac{\partial \mathbf{v}}{\partial t} + \mathbf{v} \cdot \nabla \mathbf{v}) = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f_b} \tag{3.2}$$

Here ρ is the fluid density and $\mathbf{v}$ the fluid velocity vector. The pressure is denoted by p and μ is the dynamic viscosity. Body forces acting on the fluid are represented by $\mathbf{f_b}$. This term includes gravity, surface tension, etc. Equation 3.1 is the continuity

[1] which can of course be different for the two phases.

equation and describes the conservation of mass. Equation 3.2 is the momentum equation and represents a balance between acting forces and momentum change [17, 29].

3.2 Numerics and origin of parasitic currents

This section will present some theory related to the numerical aspect of this work as the NSE are discretized in time and space. The VOF method and the numerical implementation of the surface tension will be explained. Because surface tension in VOF simulations causes parasitic currents, some important results on parasitic current magnitude will be presented.

3.2.1 Volume of Fluid Method

The Volume of Fluid (VOF) Method was presented in 1979 by Hirt and Nichols [18]. It relies on an indicator function that can take values between 0 and 1, on a fixed mesh. This phase fraction function indicates which fluid is present in a cell, or if it is a mix of both.

The flow problem is governed by the NSE mentioned in Section 3.1:

$$\nabla \cdot \mathbf{v} = 0\,, \tag{3.3}$$

$$\rho\frac{\partial \mathbf{v}}{\partial t} + \rho\mathbf{v}\cdot\nabla\mathbf{v} = -\nabla p + \mu\nabla^2\mathbf{v} + \mathbf{f_b} \tag{3.4}$$

In a VOF simulation, the term $\mathbf{f_b}$ include gravity and surface tension effects [11]. In addition to the continuity and momentum equation, the indicator function that describes the interface between the two fluids is introduced. The phase fraction "$\alpha(x, y, t)$" is advected (i.e. transported) along the velocity field. It is therefore described by an advection equation of the form:

$$\frac{\partial \alpha}{\partial t} + \mathbf{v}\nabla\alpha = 0 \tag{3.5}$$

The meaning of the phase fraction α in Equation 3.5 is:

$$\alpha = \begin{cases} 1 & \text{, drop (water)} \\ 0 & \text{, ambient fluid (D80 oil)} \\ 0 < \alpha < 1 & \text{, at the interface} \end{cases}$$

The phase fraction α can take values between 0 and 1 in the region of the interface. This means that an increase in mesh resolution will also increase the sharpness of the interface. Conservation of phase fraction (and thus mass), is therefore strongly dependent on grid resolution [11].

The density and the viscosity of the fluid in a cell of the mesh is found by using the phase fraction α [11]:

$$\begin{aligned} \rho &= \rho_{\mathrm{w}}\alpha + \rho_{\mathrm{D80}}(1-\alpha)\,, \\ \mu &= \mu_{\mathrm{w}}\alpha + \mu_{\mathrm{D80}}(1-\alpha)\,, \end{aligned} \tag{3.6}$$

where ρ and μ are the density and viscosity of the fluid in a cell, respectively. The density and viscosity of water is ρ_{w} and μ_{w} while the density and viscosity of Exxsol D80 oil is ρ_{D80} and μ_{D80}.

3.2.2 The Continuum Surface Force method

Surface tension makes it possible for drops to form in a continuous phase. The cohesive force among molecules in a liquid is the same in all directions which results in a net force equal to zero. Assuming a gas-liquid interface, molecules at the interface experience a different force, and will tend to be pulled inwards [39]. This surface energy will be smallest when taking the shape of a sphere. Surface tension is responsible for the formation of liquid drops in gas or in another liquid. The uneven force distribution is shown in Figure 3.1.

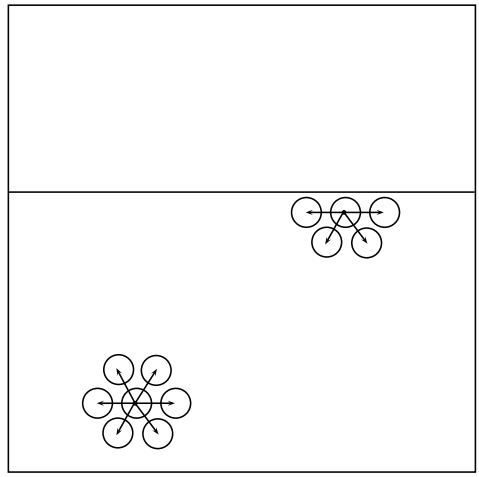


Figure 3.1: Intermolecular forces acting on a molecule in a liquid and at a gas-liquid interface. Molecules at the interface experience a different force than the molecules surrounded by liquid. To minimize the energy of the surface, molecules on the surface tend to be pulled inwards.

Surface tension effects are included in the simulations by the continuum surface force (CSF) approach developed by Brackbill et al. [8] in 1991. In addition to a gravitational term, surface tension will be included in the body forces $\mathbf{f_b}$ term in

Equation 3.4. Surface tension forces, denoted by $\mathbf{f}_\sigma$, are modelled by [8, 21]:

$$\mathbf{f}_\sigma = \sigma\kappa\nabla\alpha\,, \tag{3.7}$$

where σ is the interfacial tension between the two fluids and α is the phase fraction. The curvature κ is defined as [8]:

$$\kappa = -\nabla\cdot(\mathbf{n})\,,$$

where $\mathbf{n}$ is the normal vector to the interface. The normal vector can be computed using the gradient of the interface [17, 36]:

$$\mathbf{n} = \frac{\nabla\alpha}{|\nabla\alpha|} \tag{3.8}$$

Note: This model is only valid for fluids with constant interfacial tension. Temperature gradient or contamination by particles will introduce a surface tension gradient [11]. For the present work, a constant surface tension is assumed.

3.2.3 The origin of parasitic currents

Harvie et al. [16] have quantified the magnitude of parasitic currents in different common physical systems. They also provide an explanation for the origin of parasitic currents which will be briefly explained in this section.

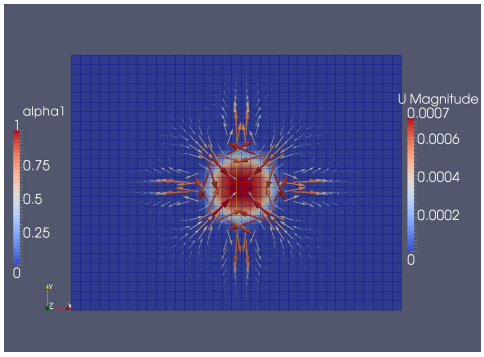


Figure 3.2: Screenshot showing the velocity field of parasitic currents in a zero gravity simulation. Screenshot is taken at $t = 0.1$ s for a drop with the viscosity of water and a surface tension of $\sigma = 0.001\mathrm{N\,m^{-1}}$. The drop diameter is $d = 500$ µm and the ambient fluid is Exxsol D80 oil.

The surface tension force $\mathbf{f}_\sigma$ in Equation 3.7 is problematic to implement in VOF methods [16], so the contribution is rewritten as a volume force [16]:

$$\mathbf{f}_\sigma = \frac{1}{\mathrm{We}}\kappa\mathbf{n}, \tag{3.9}$$

where the curvature is denoted by κ, the normal vector to the interface by $\mathbf{n}$, and the Weber number We is defined as:

$$\mathrm{We} = \frac{\rho v^2 d_d}{\sigma},$$

with ρ the density of the ambient fluid, v the relative velocity at the drop interface, d_d the drop diameter, and σ the surface tension.

The contribution of the surface tension force in Equation 3.9 and for an "ideal" VOF method will be studied. Because we are in a 2D case, we use the cylinder coordinates (r, θ) for the phase fraction function α. The phase fraction is here only a function of r, as α is equal to one for $r \ll \frac{d_d}{2}$ and zero for $r \gg \frac{d_d}{2}$. In the region $r \approx \frac{d_d}{2}$, α varies between one and zero. The normal vector $\mathbf{n}$ points radially inwards, and the curvature κ is equal to $1/r$. Thus, the surface tension force is a vector pointing radially inwards, with a magnitude only depending on r.

Such as force is irrotational and can therefore be represented as the gradient of a scalar quantity [17]. Looking back at the momentum equation, in a case where there is zero gravity:

$$\rho \frac{\partial \mathbf{v}}{\partial t} + \rho \mathbf{v} \cdot \nabla \mathbf{v} = -\nabla p + \mu \nabla^2 \mathbf{v} + \mathbf{f}_\sigma \tag{3.10}$$

One can see that all velocity dependent terms are zero, as we are considering an "ideal" VOF for a drop in equilibrium. The pressure gradient term will therefore precisely balance the term due to surface tension.

In a real VOF implementation however, the numerical discretization of the phase fraction field over computational cells results in the normal vector $\mathbf{n}$ pointing not precisely in the radial direction, as well as the curvature κ varying slightly with θ. This effect induces an "erroneous" rotational contribution to the surface tension component in Equation 3.10. As seen previously, the irrotational part will balance the pressure gradient, but now velocities must compensate for the rotational part. This is the reason velocities appear in VOF simulations, even in a zero gravity environment.

3.2.4 Magnitude of parasitic currents

Parasitic currents have been studied in the past and correlations have been deduced. However, one should be careful when comparing magnitudes of such currents, as different solvers were used by the authors. It is also difficult to give a systematic expression for the amplitude of parasitic currents. They often fluctuate with time because the interface is constantly changing [33].

Lafaurie et al. [24] showed that parasitic currents increase in magnitude with the surface tension. The authors performed a dimensional analysis and suggest the

maximum velocity around a bubble with zero gravity scales as [24]:

$$u_p = K\frac{\sigma}{\mu},\ K \simeq 0.01\,, \tag{3.11}$$

with u_p the parasitic current magnitude, σ the surface tension, and μ the dynamic viscosity of the drop.

Laufaurie et al. considered two phases with equal densities and equal viscosities, both equal to one. This relation was tested for different drop diameters, which are not specified by the authors. The measured slopes ranged from 0.009 to 0.05 with most points close to $K = 0.01$. This scatter of date is caused by a fluctuation in time of the parasitic current magnitude.

Harvie et al. studied quantitatively the magnitude of parasitic currents. Their results are in agreement with the coarse correlation of Lafaurie et al., in equation 3.11. Harvie et al. performed an order of magnitude analysis on the NSE equations with the CSF approach included. They compared in turn balances between viscous and surface tension terms, surface tension and inertial advection, and finally the balance between inertial transient terms and surface tension terms. The parasitic currents deduced from the order of magnitude are denoted respectively as U_V, U_A, and U_T[2] [16]:

$$U_V = \frac{2\max(\frac{\rho_d}{\rho_c},1)}{\mathrm{Ca}(1+\frac{\rho_d}{\rho_c})\min(\frac{\mu_d}{\mu_c},1)},\quad U_A = \sqrt{\frac{2}{\mathrm{We}(1+\frac{\rho_d}{\rho_c})h}},\quad U_T = \frac{2t_m}{\mathrm{We}(1+\frac{\rho_d}{\rho_c})h^2} \tag{3.12}$$

In Equation 3.12, h is the mesh size and t_m is the maximum time that any particular mesh cell contains an interface region. ρ_d and μ_d is the density and dynamic viscosity of the drop, respectively. ρ_c and μ_c is the density and dynamic viscosity of the ambient fluid, respectively. The dimensionless Capillary number Ca and the Weber number We are defined as:

$$\mathrm{Ca} = \frac{U\mu_c}{\sigma}\,,\quad \mathrm{We} = \frac{\rho_c U^2 d}{\sigma}\,, \tag{3.13}$$

where U is the relative velocity of the fluids, μ_c the dynamic viscosity of the continuous phase, ρ_c the density of the ambient fluid, d the drop diameter, and σ the surface tension.

Harvie et al. used the conservative assumption to argue that the erroneous surface tension force will be limited by the smallest velocity term, thus resulting in the following correlation for the maximum magnitude of parasitic currents [16]:

$$U_p = \min(a_T U_T, a_A U_A, a_V U_V) \tag{3.14}$$

In Equation 3.14, a_T, a_A, and a_T are constants specific to a particular numerical scheme.

[2] "V" denotes viscous, "A" denotes advection, and "T" denotes inertial transient

3.3 Theory on terminal velocity

In a two phase flow, gravity will act differently on fluids with different densities. Considering a drop of a fluid denser than the surrounding fluid, it will be accelerated by gravity until the drag force is equal to the gravitational force: the terminal velocity of the drop has been reached. This section will present some underlying theory on terminal velocity as well as equations predicting the terminal velocity. Two models are used to describe the terminal velocity of a drop. The Hadamard-Rybczynski assumes the drop is perfectly spherical, while the predictions of the Stokes', Intermediate, and Newton's law assume the drop is solid and spherical.

3.3.1 The Hadamard-Rybczynski equation

An analytic solution was derived for fluid spheres in a surrounding flow of infinite extent. The surface tension of the fluid-fluid interface σ is assumed to be constant, i.e without any contamination of surface-active particles. In the derivation of Equation 3.15, it is also assumed that the Reynolds number of the sphere and the surrounding fluid are small, as the flow is assumed to be laminar. The upper limit of applicability of these equations is thought of as $Re \simeq 1$ [23]. The Hadamard-Rybczynski solution reads [9]:

$$u_t = \frac{2}{3}\frac{r^2 g(\rho_d - \rho)}{\mu}\frac{\mu + \mu_d}{2\mu + 3\mu_d}\,, \tag{3.15}$$

where r is the drop radius, g the gravitational acceleration, and ρ (ρ_d) and μ (μ_d) the density and viscosity of the surrounding fluid (drop).

For a rigid solid sphere, equations for terminal velocity for different velocity regimes have been derived and will be presented in the next section. The result for Stokes' law can be derived by letting $\mu_d \to \infty$, i.e. assuming a highly viscous drop in Equation 3.15.

Note: Small drops tend to obey Stokes' law experimentally rather than the corresponding Hadamard-Rybczynski result [9]. This result may be attributed to the fact that internal circulation is absent for small drops, due to surface contaminants [9]. More explanations can be found in *Drops, Bubbles, and Particles*, Chapter 3, Section C. by Clift et al. [9].

3.3.2 Terminal velocity for a rigid sphere

The previous section presented the theoretical laws derived for a spherical liquid drop. This section will present results for solid spherical particles.

Drag coefficient

In steady flows, the drag force acting on an object, for example a falling particle, is given by [31]:

$$F_D = \frac{C_D A_d \rho u^2}{2}, \tag{3.16}$$

where F_D is the drag force, C_D the drag coefficient, A_d the projected spherical particle area in the direction of motion, ρ the surrounding fluid's density, and u the relative velocity between the particle and the surrounding fluid.

The drag coefficient C_D is determined by experimental studies done with solid spheres of different diameter, falling in fluids with different viscosities. The drag coefficient is dependent on the geometry of the object falling (in this work, it is a sphere) and the velocity of the surrounding fluid u. This results in an experimental curve, plotting the drag coefficient as a function of the Reynolds number. The fall velocity of the sphere is directly linked to the diameter of the sphere (and thus the particle density), and the viscosity of the ambient fluid.

Terminal velocity

The terminal velocity u_t of a spherical rigid particle is given by [31]:

$$u_t = \sqrt{\frac{4gd(\rho_d - \rho)}{3\rho C_D}}, \tag{3.17}$$

with g the acceleration of gravity, d the drop diameter, C_D the drag coefficient, ρ the density of the ambient fluid, and ρ_d the particle density. The drag coefficient is dependent on the velocity regime, which can be expressed by the Reynolds number Re_d of the flow around the particle:

$$Re_d = \frac{\rho_d u_t d}{\mu} \tag{3.18}$$

where ρ_d is the density of the spherical particle, u_t the terminal settling velocity of the particle, d the particle diameter, and μ the dynamic viscosity of the surrounding fluid. Different Reynold regimes will yield different drag coefficients which will yield in turn different settling velocities. The predicted terminal velocity by the Stokes', intermediate, and Newton's laws are formulas which fit the experimental data.

At low Reynolds number, Stokes' law reads [31]:

$$u_t = \tfrac{gd^2(\rho_d - \rho)}{18\mu}, \quad 0.0001 < Re_d < 2 \tag{3.19}$$

Stokes' law applies for small particle diameters and very viscous fluids. For medium sized droplet diameters, the Intermediate law applies [31]:

$$u_t = \tfrac{0.153 g^{0.71} d^{1.14} (\rho_d - \rho)^{0.7}}{\rho^{0.29} \mu^{0.43}}, \quad 2 < Re_d < 500 \tag{3.20}$$

For larger velocities, Newton's law applies [31]:

$$u_t = 1.74\sqrt{\frac{gd(\rho_d - \rho)}{\rho}}, \quad 500 < Re_d < 200\,000 \tag{3.21}$$

In Equations 3.19, 3.20, and 3.21, u_t is the terminal velocity, g the acceleration of gravity, d the drop diameter, ρ_d and ρ the density of the particle and the ambient fluid, respectively, μ the dynamic viscosity of the ambient fluid, and Re_d the Reynolds number.

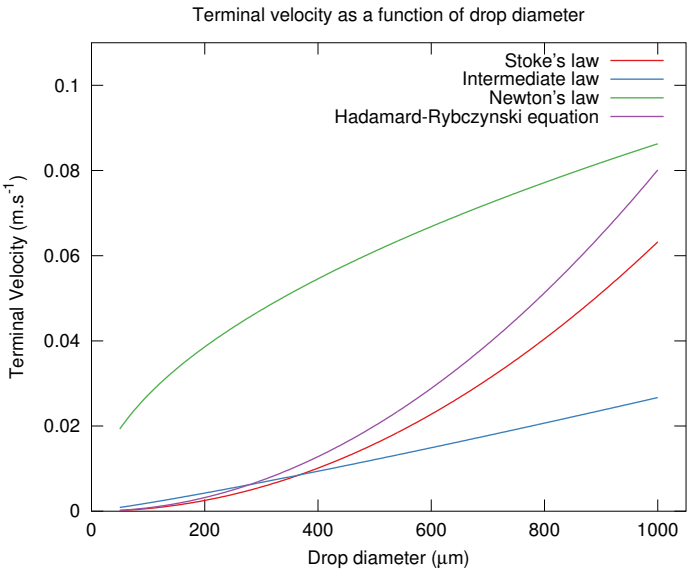


Figure 3.3: Plot of the theoretical laws for the terminal velocity of a rigid or non-rigid sphere. Drop diameter is $d = 500$ µm, the drop viscosity is $\mu_d = 1.0 \cdot 10^{-3}$ Pa s, the ambient fluid has the viscosity of Exxsol D80 oil $\mu_{\mathrm{D80}} = 1.7 \cdot 10^{-3}$ Pa s and the surface tension is $\sigma = 0.005$ N m^{-1}.

Figure 3.3 shows the terminal velocity as a function of drop diameter. All equations mentioned here assume a spherical shape of the drop, but the Hadamard-Rybczynski equation takes into account internal circulation, as the viscosities of both phases are present in Equation 3.15. Figure 3.3 shows that the predicted terminal velocity is higher with the Hadamard-Rybczynski equation. This result can be explained by internal circulation, as this phenomenon will decrease the velocity gradient at the interface, thus reducing the drag force [23]. This phenomenon has greater influence for larger drop, which is also seen in Figure 3.3.

Falling liquid drops in reality will be horizontal ellipsoids. This shape has a bigger projected area in the fall direction, thus yielding a higher drag force. This has

been experimentally observed by Licht and Narasimhamurty [26] and will therefore result in some deviation from the theoretical predictions for perfectly spherical particles.

Note: Depending on the drop regime, the drop can for example be skirted or a spherical cap (cf. Figure 2.1). In such regimes, the laws mentioned in this section do not apply.

CHAPTER 4

OpenFoam: case and post-processing

This chapter will present the parameters specific to the simulations for this work, after a brief presentation of the CFD tool OpenFoam. Mesh generation, boundary conditions, and initial conditions will be presented. Finally, the procedure for extracting the useful information; drop velocity, drop position, and drop area; will be explained.

4.1 Presentation of OpenFoam

OpenFoam[1] is an open source CFD toolbox programmed in C++ that includes numerical solvers for important governing differential equations in a very wide range of fields. Most solvers are for Computational Fluid Dynamics: *interFoam* for multiphase problems and *chemFoam* for combustion problems are only two of numerous available solvers. OpenFoam can also be used to simulate cases in Electromagnetics and even Finance. The code is open source, meaning that anyone can get access to the underlying code and even change it to better fit the problem. OpenFoam is compiled on Linux machines[2] and supports parallel computing which makes it very attractive for high performance computations on clusters.

Training to use OpenFoam comes mainly from running tutorial cases provided with the toolbox, reading the user guide, or through discussions on Internet with other users. For example, the website http://www.cfd-online.com has one of the biggest community in CFD, and is a very useful source of information for OpenFoam users.

4.1.1 Pre-processing

There is no graphical interface when using OpenFoam, as all the information related to a simulation is written in text files placed in different folders. These files are then read by the solver when the simulation is run. The different elements of a case are specified in different folders:

[1] Open Source Field Operations and Manipulations
[2] OpenFoam is also available on Windows computer

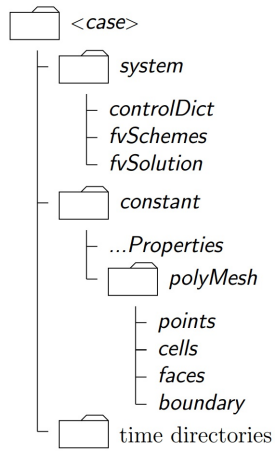


Figure 4.1: Structure of an OpenFoam case [4].

- **system** directory: in this folder, control parameters such as start and end time, time step, and write frequency can be specified in the *controlDict* file. Options about the discretization scheme may be set in the *fvSchemes*, while *fvSolution* lets the user choose equation solvers and tolerance levels [4].
- **constant** directory: this directory contains fluid properties, constant parameters (gravity for example), and information about the mesh, presented in text files with all coordinates of vertices. The mesh can be generated by using the *blockMesh* function [4], which will generate the mesh for simple geometries. Another option is to import the mesh from other CFD tools such as Fluent or ANSYS. Boundary conditions are specified in this folder, by writing values manually or by using OpenFoam functions, like *turbulentInlet* for example, for turbulent velocities at the inlet of the case.
- **time** directories: these folders will appear after a simulation has been run. They will contain information about pressure, velocities, and other variables at all grid points. Each time step will have its own **time** folder, according to the 'writeFrequency' specified in the *controlDict* file. The **0** folder will include the initial and boundary conditions of the problem.

4.1.2 Solving

Once the case has been set up, the simulation can be run by using an OpenFoam solver. Depending on the problem, the user can choose among several solvers. The solver is called on the Linux command line and the **time** folders will be created for each time step: they will contain the information about pressure, velocities, phase fraction, chemical species fraction, and any other field of interest. They will be written in text files for each grid point.

4.1.3 Post-processing

To be able to read the data after the simulation has been run, the user can use the open source software ParaView, or convert to another format so the user can post-process with another CFD tool. ParaView makes it possible to visualize the fields for pressure, velocity, and phase fraction for example, with nice animations and colors. For this work, ParaView 3.12.0 is used.

Another option is to use the *sample* function of OpenFoam: it will run through the **time** directories and copy the values of interest along lines or at points given by the user. The extracted data can be loaded using any numerical computing language such as Matlab, Octave, Python, or Gnuplot.

The last main option is to use the *swak4Foam*[3] toolbox. This library is independently developed and require installation and compilation on the computer. The *swak4Foam* toolbox can write down information of interest upon runtime. The toolbox allows the user to put some conditions on the data extracted and also specify expressions involving the fields [3].

4.2 The OpenFoam case

This section will present details and choice of parameters specific to the OpenFoam case for the fall of the drop. Mesh generation, boundary and initial conditions, and different aspects specific to the CFD tool used will be presented.

In regards of the CPU power available, a regular laptop with a 2.26 GHz Intel Core 2 Duo P8400 processor, the model chosen will be two-dimensional (2D). Temperature is fixed at 298 K and we neglect heat exchange. This is a reasonable assumption, with respect of the velocities we obtain for a laminar flow [25]. The flow considered here is two-dimensional, unsteady, incompressible, laminar, isothermal, and the Volume of Fluid method (VOF) , implemented in the *interFoam* solver of OpenFoam 2.1.1 is used.

4.2.1 Grid generation

For simple geometries, the mesh can be generated by using the *blockMesh* function. The coordinates of the vertices describing the geometry and the number of cells are entered [4].

The *refineMesh* function has then been used to increase the resolution in the center part of the mesh, as seen in Figure 4.2 or on the screenshot taken from ParaView in Figure 4.3.

4.2.2 Choice of solver: *interFoam*

Simulations were performed by using the *interFoam* solver from the CFD package OpenFoam. This solver is suitable for multiphase problems of incompressible flu-

[3]Swiss Army Knife for Foam

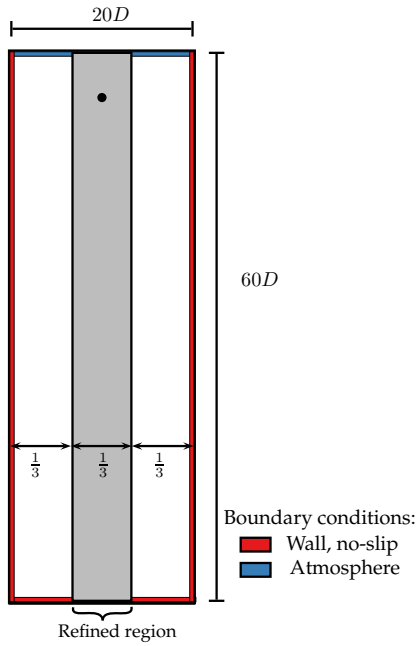


Figure 4.2: Geometry of the two-dimensional container simulating the fall of the drop. D is the diameter of the drop.

ids [4]. The flow is considered to be laminar, as viscosities are high and velocities low [35]. We based our case on the "damBreak" tutorial [4] which is included in the OpenFoam toolbox after compiling. The reader is referred to look at this case for details about numerical schemes, solvers, relaxation factors, etc.

4.2.3 Boundary and initial conditions

Velocity and pressure field

The fall of the drop is considered in a quiescent flow. Hence, the initial velocity field is set to zero in all cells. The internal pressure field is set to zero, with a reference point in the lower right corner of the mesh. This will result in a hydrostatic pressure field starting at zero at the bottom of the container and decreasing with negative pressure upwards due to less hydrostatic pressure.

In order to reduce computing time and reach terminal velocity faster, an initial velocity field downwards has been set for all cells within the drop, thus creating an "initial push". This is done for some simulations and rapid testing showed this trick was valid, as terminal velocity was unchanged while computing time could be reduced.

ρ_w [kg m^{-3}]	ρ_{D80} [kg m^{-3}]	ν_w [m^2 s^{-1}]	ν_{D80} [m^2 s^{-1}]	σ [N m^{-1}]
998	798	$1.004 \cdot 10^{-6}$	$2.16 \cdot 10^{-6}$	0.03444

Table 4.1: Thermodynamic properties of water and Exxsol D80 oil. σ is the water-oil surface tension, measured by Danielson [12], ρ the density of the fluid, and ν the kinematic viscosity of the fluid. All the data are given at room temperature $T = 298$ K.

Phase fraction

The phase fraction is denoted by "alpha1", or α_1. For the continuous fluid, the Exxsol D80 oil, the phase fraction is zero, while a phase fraction of one corresponds to water. This can be seen in Figure 4.3. The thermodynamic properties of the fluids are shown in Table 4.1. The surface tension between Exxsol D80 oil and water has been measured by Thomas J. Danielson [12] and viscosities were found in the product data sheet of the Exxsol D80 oil [5].

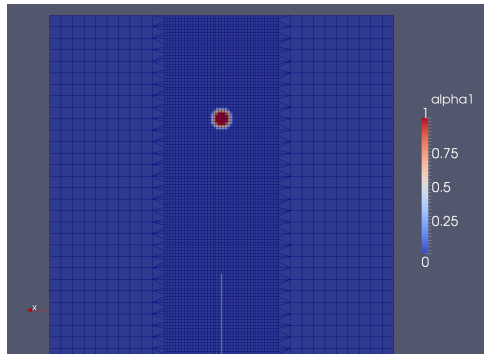


Figure 4.3: Screenshot of the initial phase fraction field in ParaView.

The *setFields* function was used to initialize the phase fraction field by specifying all cells to contain Exxsol D80 oil, i.e. $\alpha_1 = 0$. The *swak4Foam* toolbox with the function *funkySetFields* [3] was used to create the phase fraction field for the water drop, as given in Code 4.1. This line of code takes all cells within a specified radius from the center of the drop and changes the phase fraction to 1 by using the equation:

$$(x - x_{\text{drop}})^2 + (y - y_{\text{drop}})^2 \leq \text{radius}^2 \,,$$

where x_{drop} and y_{drop} are the coordinates of the center of the drop.

4.2.4 Time control with the *controlDict* file

In this file, time control parameters are specified. One can specify the write frequency of data, the time step of the simulation, and the end time. This file will

```
funkySetFields -field alpha1 -expression 1 -time 0 -keepPatches -
    condition "pow(pos().x-(20*0.000500/2),2)+pow(pos().y
    -(0.9*60*0.000500),2)<=pow((0.000500/2),2)"
```

Code 4.1: Initialization of the phase fraction field with *funkySetField*. Drop diameter is 500 µm. The drop is located at 90% of the container's height and at 50% of the container's width. The height and width of the container are 60 and 20 times the drop diameter, respectively.

also write data of interest, (the velocity for example) for post-processing during the run time.

The toolbox used for post-processing is the *swak4Foam* toolbox. It will write data to a file which can be loaded for post-processing. The code for finding the drop velocities, the drop center, and the droplet area will be presented in the next sections.

Time step

The Courant number Co, also called the Courant-Friedrichs-Lewy condition (CFL) is defined as:

$$\mathrm{Co} = \frac{u_x \Delta t}{\Delta x} + \frac{u_y \Delta t}{\Delta y} < \mathrm{Co}_{\max}\,, \tag{4.1}$$

with u_x and u_y the horizontal and vertical velocity components, respectively, Δt the time step of the simulation, Δx and Δy the horizontal and vertical grid spacing, and $\mathrm{Co}_{\max}$ the maximum Courant number.

The Courant number restricts the possibility of fluid moving through more than one cell between one time step of the simulation. This is because the computation of fluxes assumes the cells are adjacent [18].

In the **controlDict** file, the time step has been set to be variable, by switching the "adjustableRunTime" to true. This choice has been made to save computation time: to ensure stability, the Courant number in Equation 4.1 is required to be less than a certain value $\mathrm{Co}_{\max}$, depending on the numerical methods involved. However, a very small Courant number will waste computing time. Based on the *damBreak* tutorial [4], the maximum Courant number has been set to $\mathrm{Co}_{\max} = 0.5$.

4.3 Post-processing

Three key parameters are of interest for this work: the drop velocity, the center of gravity of the drop, and the drop area. Information is written down while the simulation is run using *swak4foam*. Octave, a third party software, is used to plot and do further calculations on the data written down at each time step.

4.3.1 Data from swak4Foam

The *swak4Foam* toolbox allows the user to specify expressions involving the fields, evaluate them, and save them to a file for each time step of the simulation. Some details related to saving the drop velocity and drop coordinates follows.

Drop velocity

The velocity of the drop is calculated in *swak4Foam* by using the following formula:

$$\overline{U_d} = \frac{1}{\Omega} \int_{\Omega} U \, \mathrm{d}\Omega \,,$$

where Ω is the domain defined by the drop, $\overline{U_d}$ the extracted average velocity, and U the velocity magnitude in a cell of the domain Ω. This formula is calculated by *swak4Foam* and is saved for each time step. The domain of the drop is defined by cells for which $\alpha > 0.5$.

The code for finding the drop velocity, which is included in the *controlDict* file, is in Appendix A, Code A.1. Said simply, this code will look at the cells defining the droplet in the grid, add their downwards velocities and divide by the volume of the cells used. This code writes down the averaged velocity of the drop for each time step in a file.

Horizontal and vertical coordinates

Similarly, *swak4Foam* writes the maximum and minimum coordinates of the drop. As before, the domain of the drop is defined for cells for which $\alpha > 0.5$. The *swak4Foam* code is shown in Appendix A, Code A.2.

4.3.2 Post-processing with Octave

Post-processing has been performed using GNU Octave, version 3.2.4, which is a free and open source software for numerical computations. Syntax and functionalities are very similar to Matlab. Data saved with *swak4Foam* has been loaded into Octave scripts.

Terminal and parasitic velocities

The terminal velocity is found by using the Octave function shown in Appendix A, Code A.3. The data from the simulation is loaded and fitted by a 4^{th} order polynome. The slope is then considered, and points that have a slope lower than a tolerance value, i.e. the points where the velocity is not changing are kept for averaging. The maximum slope tolerance is 0.001. Figure 4.4 shows the different steps just mentioned.

This method required systematic check that the extracted value was correct. To ease the procedure in some cases, the simulation was interrupted a while after the terminal velocity was reached. This was done by visual inspection. An arithmetic mean was performed on the last points to obtain a terminal velocity. Accuracy on

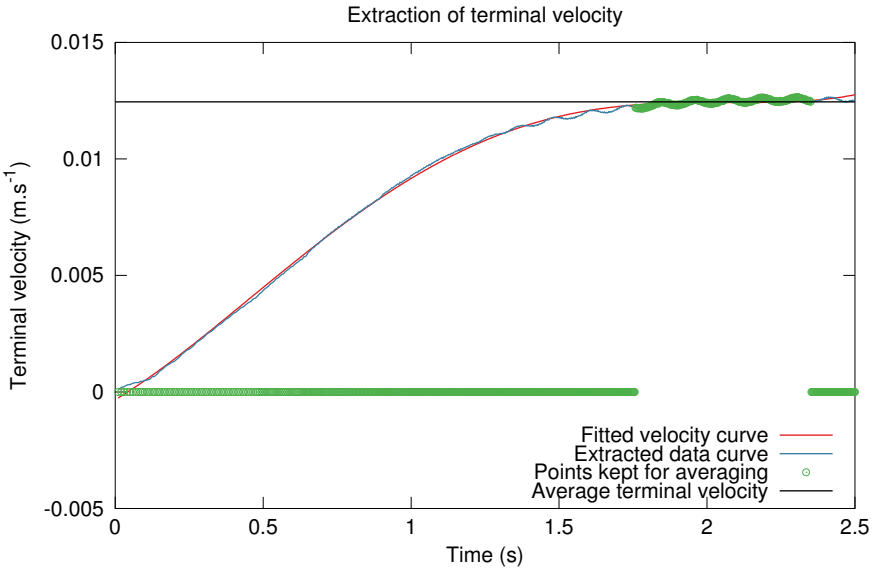


Figure 4.4: Plot showing the procedure in Octave to extract the drop terminal velocity from the data written by *swak4Foam*. The data is fitted by a 4^{th} order polynome. Non-zero points in green have a slope close to zero. Non-zero points are kept for averaging.

the extracted terminal velocity is unchanged, if not better.

The parasitic current magnitude was extracted using a zero gravity simulation. The drop should be in equilibrium, but as explained in Chapter 3, Section 3.2, parasitic currents appear. In such a simulation, the velocities in cells within the drop will fluctuate before settling into a more constant profile. The initial fluctuations are explained by the relaxation of the drop from an arbitrary initial position to a circular shape. The drop is "snapping" into the mesh, as seen in Figure 4.6. After these oscillations have settled, velocities remain due to parasitic currents. Figure 4.5 shows the initial oscillations and the points kept for averaging to determine the parasitic current magnitude.

Note: The choice of using velocities after the initial relaxation to evaluate the parasitic current magnitude was somewhat arbitrary and this choice is disputable. The initial velocities are as relevant as the one coming later. Both will affect the quality of the results. However, we thought keeping the points after settling would be better with regards to the mesh. Small and large drops will have different relaxations and the objective was to remove this component when evaluating parasitic current magnitude.

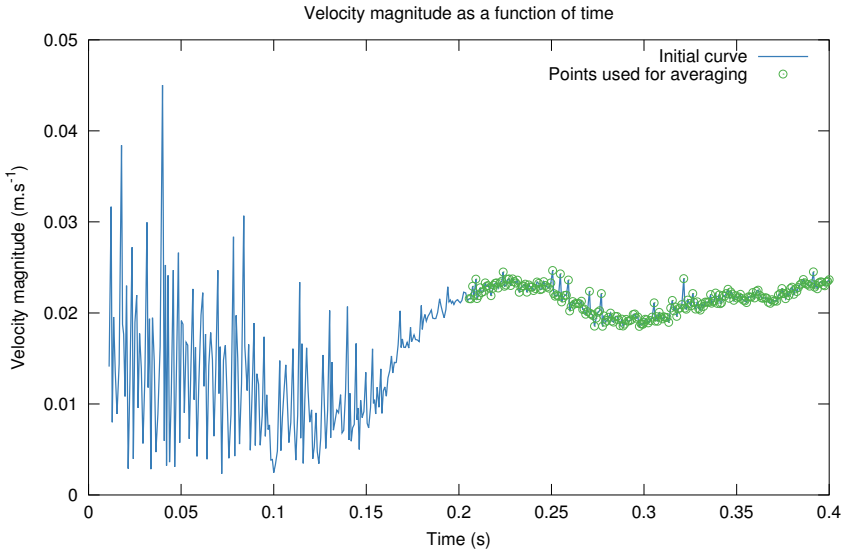


Figure 4.5: Plot showing the procedure to extract the parasitic current magnitude. In a zero gravity environment, the simulation is run until the velocities have settled. The first oscillations are due to the drop relaxation. The velocities remaining after are also due to parasitic currents. An average of the settled points is used as a measure of the magnitude of parasitic currents. The water drop has diameter $d = 500$ µm, the surface tension is $\sigma = 0.01\ \mathrm{N\,m^{-1}}$, and the ambient fluid is Exxsol D80 oil.

Drop area

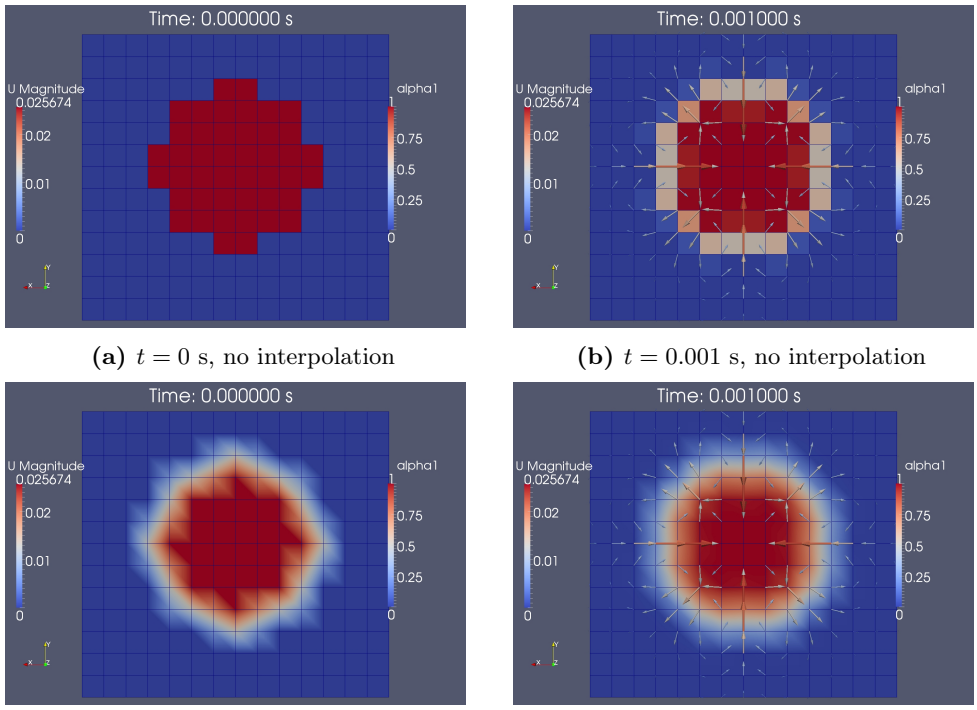
Tracking the evolution of the area of the droplet is important to check conservation of mass. Even if it is not a parameter essential for this work, some information on finding the drop area is given nonetheless. The maximum and minimum coordinates of the drop are written by the *swak4Foam* Code A.2 of Appendix A. Taking the difference yields the major and minor axis of the drop shaped as an ellipse. This is shown in Figure 4.7. The drop area can be found using the formula of the area of an ellipse:

$$A = \pi a b\,,$$

with a and b the minor and major axis, respectively.

Drop center of gravity

The center of gravity can be used to plot the path of the drop when falling. Using the minimum and maximum coordinates in the horizontal and vertical direction, one can find the center of gravity of the drop. Dividing the horizontal and vertical diameters by two yields the coordinate of the center of the drop.



(a) $t = 0$ s, no interpolation

(b) $t = 0.001$ s, no interpolation

(c) $t = 0$ s, with interpolation

(d) $t = 0.001$ s, with interpolation

Figure 4.6: Screenshot showing the relaxation of the drop to a circular shape in a zero gravity simulation. The drop changes shape because the initial drop interface is not smooth: at the interface, phase fraction is either one or zero at $t = 0$ s. At the next time step, cells can obtain values between zero and one. The two lower screenshots are from the same simulation and show how ParaView interpolates the phase fraction field for a better visualization experience. The water drop has diameter $d = 500$ µm, the surface tension is $\sigma = 0.01\ \mathrm{N\,m^{-1}}$, and the ambient fluid is Exxsol D80 oil.

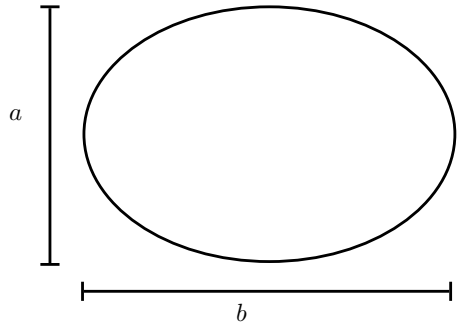


Figure 4.7: Horizontal and vertical diameter of the drop

CHAPTER 5

Results and discussion

This chapter will present the main results of this work. First, the verification and validation of the models will be presented. An evaluation of the best settings for the template case will be performed with a preliminary analysis. The pressure field is checked against the prediction of the Young-Laplace equation. Afterwards, a case validation is considered, comparing results from Lekhlifi et al. and *interFoam*. The parasitic current magnitude is then studied by comparing results from Lafaurie et al., Harvie et al., and the results from the *interFoam* solver.

5.1 Preliminary analysis

This section will look at different parameters of the numerical simulations and check their effect. This preliminary analysis is done to construct a template case which will be used to produce the final results. For this purpose, several simulations have been set up where the container width, the definition of the drop domain, and the cell size have been changed, keeping all other parameters constant.

Note: All simulations for the preliminary analysis are run with the thermodynamic properties of water and Exxsol D80 oil, except from the surface tension and the drop viscosity. The physical properties of the fluids are specified in Table 5.1. Drop diameter is $d = 500$ µm throughout the preliminary analysis. This choice was made to avoid parasitic currents and have good results, hence giving a good basis for comparing and choosing container size, isosurface threshold, and cell size.

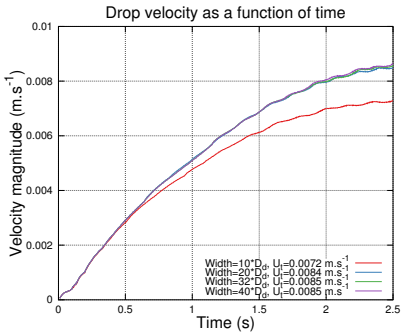
d [µm]	ρ_d [kg m^{-3}]	ρ_c [kg m^{-3}]	ν_d [m^2 s^{-1}]	ν_c [m^2 s^{-1}]	σ [N m^{-1}]
500	998	798	$1.004 \cdot 10^{-6}$	$96 \cdot 10^{-6}$	0.005

Table 5.1: Thermodynamic and physical properties of the fluids in the preliminary analysis of the simulations. The drop diameter is denotes by d, σ is the surface tension, and ν is the kinematic viscosity. Subscript "d" denotes the drop and subscript "c" denotes the continuous ambient phase.

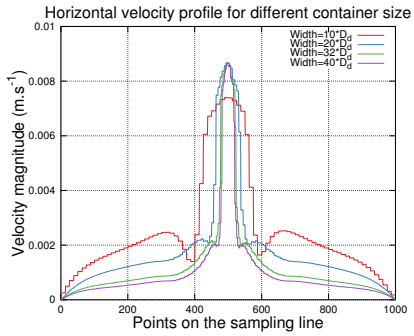
5.1.1 Container width and wall effects

Wall effects were studied first. The goal is to simulate a flow of infinite extent because it is an assumption for the underlying theory. Furthermore, this is a common assumption in experimental setups and this is also the case for the experimental setup at NTNU. One wishes negligible wall effects. This was done by simulating the fall of a 500 µm drop with parameters shown in Table 5.1 and considering several container widths.

The terminal velocity is extracted as mentioned in Chapter 4, Section 4.3. The velocity profiles are plotted using the *sample* function of OpenFoam, by extracting the values along a horizontal line passing through the center of the drop, when terminal velocity is reached. Results are shown in Figure 5.1.



(a) Drop velocity as a function of time.



(b) Velocity profile along a horizontal line passing through the drop center. The velocity peak in the center corresponds to the downwards velocity of the drop.

Figure 5.1: Plot of terminal velocity and velocity profile for different container sizes. The velocity profiles were obtained by sampling velocities along a horizontal line passing through the drop center when terminal velocity is reached, at $t = 5$ s. The simulations are for a water drop in Exxsol D80 oil. The drop diameter is $d = 500$ µm and the surface tension is $\sigma = 0.005$ N m^{-1}.

Figure 5.1 shows that the smallest container generates some wall effects as the velocity profile is not constant far away from the column where the drop falls. In this case, the wake of the drop affects the terminal velocity. Simulations will be run with a container twenty times larger than the drop diameter, as a trade-off between computing time and the quality of the results. One could use "slip" boundary conditions at the side walls, to potentially reduce wall effects. This option has not been tried in this work.

5.1.2 Effect of local refinement

The center part of the container can be refined, as shown in the screenshot in Figure 4.3. In this area, one cell is divided into four smaller cells. The effect of the

local refinement is evaluated by considering the fall of a drop with the parameters shown in Table 5.1, for a mesh with different resolutions, and with or without local refinement.

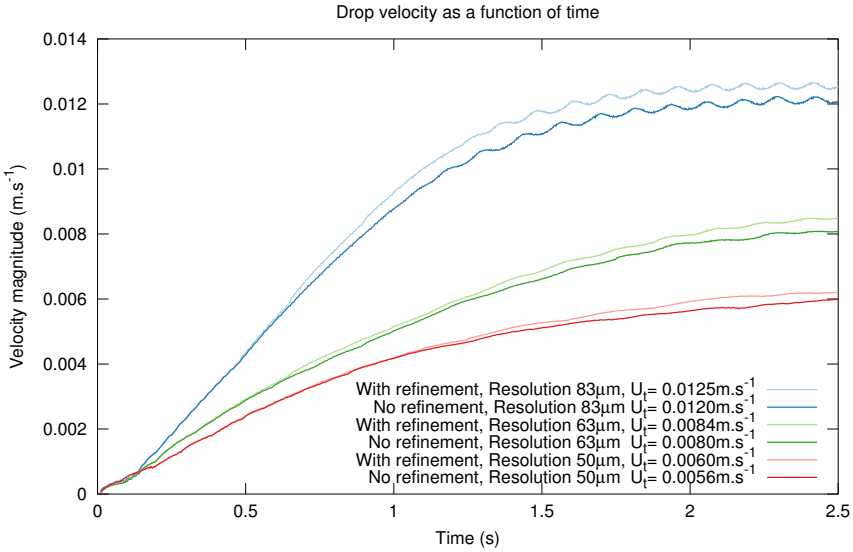


Figure 5.2: Plot on the effect of local refinement on terminal velocity. The simulations are for a water drop in Exxsol D80 oil. The drop diameter is $d = 500$ µm and the surface tension is $\sigma = 0.005$ N m^{-1}. Velocity variations are due to parasitic currents making the drop oscillate from right to left. Terminal velocities are not the same because of the mesh resolution.

Figure 5.2 shows that the local refinement in the center part of the container has a small effect on the final solution. However, the gain in computing time while keeping a good mesh resolution is drastically increased. Therefore, this trick will be used throughout this work.

Note: In Figure 5.2, terminal velocities of the drop are not the same, even if the drop has the same radius. This is due to a difference in the cell sizes of the mesh. This effect will be presented in the section dedicated to the study of the mesh resolution. The apparition of oscillations in the terminal velocity is due to parasitic currents and will be explained in Section 5.5.

5.1.3 Surface tracking by isosurface

Isosurfaces are used to select cells for extracting the averaged drop velocity, as explained in Chapter 4. In Appendix A, Code A.1, the value of "*thres*" has been modified. Initially, a value of $\alpha = 0.5$ was used, but Figure 5.3 shows that this choice has a negligible effect on the terminal velocity. This can be explained eas-

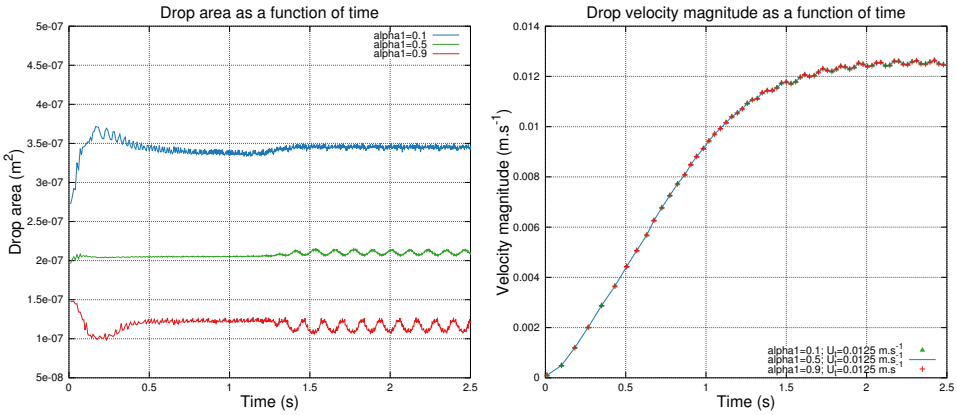


Figure 5.3: The effect of the isosurface on finding the terminal velocity. The simulations are for a water drop in Exxsol D80 oil. The drop diameter is $d = 500$ µm and the surface tension is $\sigma = 0.005$ N m^{-1}. The drop area is different depending on which cells are registered. Velocities are the same.

ily: at the interface, the effect of keeping or not a few cells is insignificant when performing the averaging.

One can also note that the drop areas are not the same. This is because the number of cells kept for defining the area are highest when α is smallest: as we move outwards the drop, the phase fraction gradually decreases from 1 to 0.

5.1.4 Mesh independence

This section will present the effect of varying the cell size of the mesh, keeping all other parameters constant and equal to the values of Table 5.1. Figure 5.4 shows the simulations for different cell sizes, ranging from 125 µm to 35 µm.

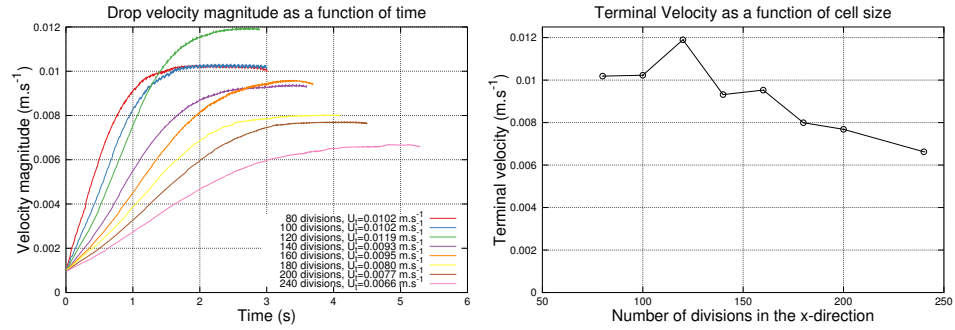


Figure 5.4: Plot of velocity and terminal velocity for an increasing mesh resolution. The simulations are for a water drop in Exxsol D80 oil. The drop diameter is $d = 500$ µm and the surface tension is $\sigma = 0.005$ N m^{-1}.

Note: Simulations have been run for different length of time and the container height has also been adjusted. This is to ensure that the drop doesn't hit the bottom of the container and that terminal velocity has been reached when the simulation ends.

Figure 5.4 shows that the terminal velocity decreases when increasing the resolution. This can be explained with results from numerical mathematics: increasing the resolution will describe the problem more accurately until the condition number of the matrix defining the problem will start increasing, thus yielding more errors on the final solution [37]. In other words, the terminal velocity will decrease when increasing the resolution until a minimal value is reached. A mesh too fine will create numerical errors and the terminal velocity will be erroneous for a grid too fine. One should also note that the first points do not follow the trend. A possible explanation is that the mesh is too coarse to describe the problem accurately.

For further studies, the mesh is chosen with 120 divisions in the center part. This corresponds to a cell size of 83.3 µm. The simulations with this resolution (20520 cells in total) last for about 15 minutes on the regular laptop computer used for this work. This resolution has been chosen to ensure an efficient work flow, even though the result of the mesh independence study would suggest using a finer grid. No explanation was found for the peak value. We chose a "the more cells, the better" approach and used the best resolution we could work with the laptop computer.

5.2 Verification of the Young-Laplace equation

The Young-Laplace equation describes the pressure difference sustained across a curved interface of two fluids due to surface tension. For a drop of diameter $d = 500$ µm and with a surface tension $\sigma = 0.001$ N m^{-1}, the predicted pressure jump Δp is [39]:

$$\Delta p = \frac{4\sigma}{d} = 8 \text{ Pa}\,,$$

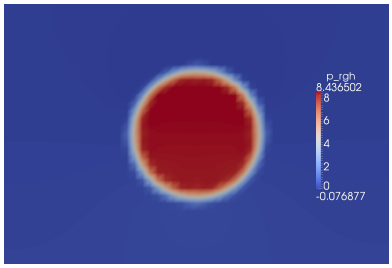
where d is the drop diameter and σ the surface tension. A three-dimensional simulation has been set up with these values to check if the pressure difference corresponds to the prediction of the Young-Laplace equation. Results can be seen in Figure 5.5.

Figure 5.5 shows that the pressure jump at the interface is very close the predicted value, with an error of 6%. This deviation can be explained by the curvature: the drop is not perfectly spherical in a simulation.

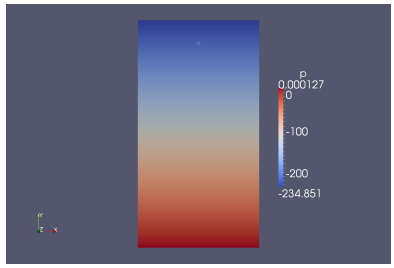
The hydrostatic pressure field can be found by using:

$$\Delta p = \rho g h\,,$$

where Δp is the pressure difference, ρ the fluid density, g the acceleration of gravity, and h the height of the fluid column. In Figure 5.5, a container of height $h = 0.03$ m



(a) p_rgh is the pressure field with the hydrostatic pressure subtracted. The measured pressure jump at the interface is $\Delta p = 8.5$ Pa.



(b) The hydrostatic pressure field. Container height is sixty times the drop diameter, i.e. $h = 0.03$ m. The pressure difference from top to bottom is $\Delta p = 234.85$Pa.

Figure 5.5: Figure showing the pressure field in the drop and the container. The simulation is three-dimensional and for a water drop with diameter $d = 500$ µm in Exxsol D80 oil. Surface tension is $\sigma = 0.001$ N m^{-1}.

is used, which yields a hydrostatic pressure difference of $\Delta p = 234.85$ Pa. This is exactly what is observed in OpenFoam.

Note: The hydrostatic pressure field in Figure 5.5 is negative because the reference point has been set in the lower right corner of the mesh. The pressure at this point is zero and will decrease with negative values when we are rising in the box.

5.3 Comparison with results of Lekhlifi et al.

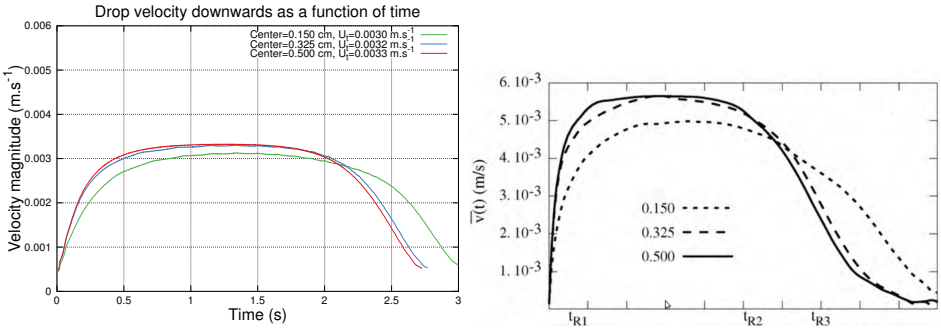
As a case validation, we wished to replicate results from Lekhlifi et al. [25]. The authors simulated a water drop falling in paraffin oil. Thermodynamic properties used for the fluids are shown in Table 5.2. The droplet diameter is $d = 2$ mm. The container size is 1 cm$\times$ 1 cm and the region is divided into 200 $\times$ 200 cells. Lekhlifi et al. do not precise the solver used for their article, but simulations with a VOF method and the CSF model was performed. This is similar to features of the *interFoam* solver used in this work [11].

Figure 5.6 shows the difference in the simulations. The terminal velocity of the drop is not the same in the two simulations. However, the general trend is the same: a smaller terminal velocity with *interFoam* yielded a longer time for the drop to reach the bottom of the container. This is why the curves from *interFoam* are slightly stretched horizontally.

Figure 5.7 shows the theoretical predictions for terminal velocity (both rigid sphere and spherical fluid) as well as the simulation results from Lekhlifi et al. and

ρ_w [kg m^{-3}]	ρ_{po} [kg m^{-3}]	μ_w [Pa s]	μ_{po} [Pa s]	σ [N m^{-1}]
1000	860	0.00106	0.0099	0.02

Table 5.2: Thermodynamic properties of water (subscript "w") and paraffin oil (subscript "po") used by Lekhlifi et al. [25]. ρ is the density of the fluid, μ the dynamic viscosity of the fluid, and σ the surface tension. All the data are given at $T = 298$ K.



(a) Results from *interFoam* **(b)** Results from Lekhlifi et al. [25]

Figure 5.6: Comparison of the velocity evolution for *interFoam* and the water-paraffin oil system of Lekhlifi et al. [25]. The three curves are for different initial center position in the container, given by the distance from the left wall. The final time of the graph from Lekhlifi et al. is $t = 2$ s. Drop diameter is $d = 2$ mm.

interFoam.

Iterations were done to find the expected terminal velocity of the water drop in paraffin oil using the model of a rigid sphere [31]. To know which regime (Stokes' or Intermediate law) is valid, one need the terminal velocity. One can not use the result from the simulation, as we do not know if it correct yet. An iteration is therefore performed, as explained in the flowchart in Figure 5.8. First an initial velocity is guessed. The Reynolds number can be calculated with this initial guess. The Reynolds number decides if we are in a Stoke or Intermediate regime. A new theoretical terminal velocity is compared to the initial guess, until the difference between the two is very small. The Octave script in Appendix A, Code A.4 shows how the theoretical terminal velocity was computed.

Figure 5.7 shows that result from *interFoam* are very close to the Stokes' law, while results from Lekhlifi et al. are closer to the Hadamard-Rybczynski relation. The terminal velocity of Lekhlifi et al. fall within a range of 20% of the predicted value of the Hadamard-Rybczynski law. The terminal velocity found with *interFoam* has an error of 7% compared to the predicted velocity of Stokes' law. Because Lekhlifi et al. do not precise the software or code used to produce their results, it is difficult to give a reason for this. A possible explanation is that their code is

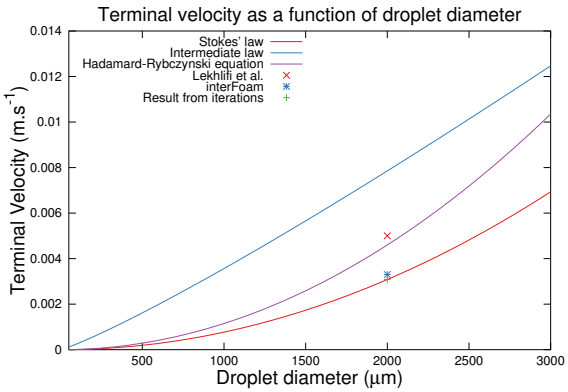


Figure 5.7: The terminal velocities obtained by Lekhlifi et al. and from *interFoam* are compared to the theoretical laws of terminal velocity. Iterations are done to find which drop velocity regime is applicable using the rigid sphere model. The theoretical value of the spherical drop model of Hadamard-Rybczynski is also plotted.

more influenced by internal recirculation in the drop than *interFoam*. Since the Hadamard-Rybczynski relation includes internal recirculation in the drop, this can explain why their results are close to the Hadamard-Rybczynski prediction.

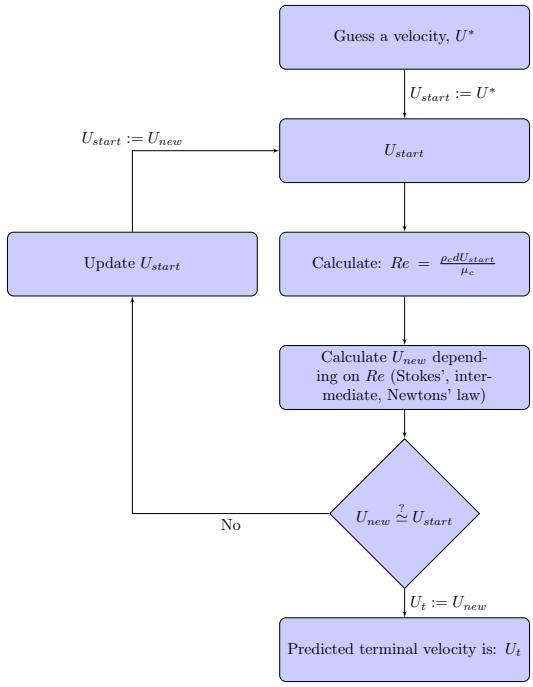


Figure 5.8: Flowchart explaining the iteration process of finding the theoretical terminal velocity of a rigid sphere. After an initial guess, the Reynolds number indicates which regime applies (Stokes', Intermediate, Newton's law). A new predicted terminal velocity can be compared to the initial velocity until convergence is achieved.

5.4 Comparison of parasitic current magnitude

In Chapter 3, Section 3.2, some correlations on parasitic current magnitude were presented. These results were proposed by Laufaurie et al. [24] and Harvie et al. [16]. The following section will compare these results.

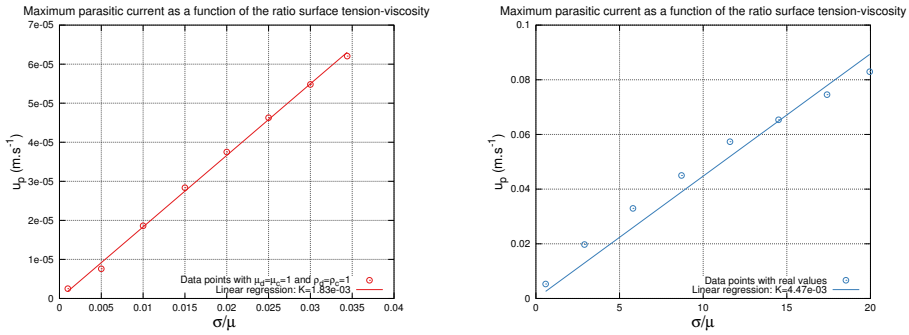
5.4.1 Correlation of Laufaurie et al.

The results of Lafaurie et al. [24] are explained in Chapter 3, Section 3.2. Lafaurie et al. looked at the parasitic current magnitude for a zero gravity simulation. They suggest that for fluids densities and viscosities all equal to one, the magnitude of parasitic current scale as:

$$u_p = K\frac{\sigma}{\mu},\ K \simeq 0.01$$

We wish to compare this prediction to simulation results of *interFoam*. Figure 5.9 shows the two results obtained with the *interFoam* solver. The same case

considered by Lafaurie et al., with all fluid densities and fluid viscosities equal to one was simulated with *interFoam*. The drop diameter considered was $d = 500$ µm. Another range is considered, with the density and viscosity of Exxsol D80 oil and water, and a drop diameter of $d = 500$ µm. A linear regression through origin is performed.



(a) Same range as Lafaurie et al. [24]: $\mu_c = \mu_d = 1$, $\rho_c = \rho_d = 1$. Drop diameter is 500 µm. Slope is $K = 1.8 \cdot 10^{-3}$.

(b) Viscosities and densities of water and Exxsol D80 oil. Surface tension is $\sigma = 0.005$ N m^{-1} and drop diameter is $d = 500$ µm. Slope is $K = 4.5 \cdot 10^{-3}$.

Figure 5.9: Plot showing the magnitude of parasitic current for different surface tensions using *interFoam*. The graph on the left tests the surface tension and viscosity range in which Lafaurie et al. operated in. The graph on the right uses the physical densities and viscosities of water and Exxsol D80 oil, with various surface tensions.

In both cases in Figure 5.9, the slope is approximately an order of magnitude smaller than the prediction of Lafaurie et al. The difference in the slopes of the two graphs, both produced by *interFoam* in Figure 5.9 may be due to the fact that the magnitude of parasitic currents are difficult to evaluate as they fluctuate in time [33].

The difference in the order of magnitude of the slopes, compared to the slopes obtained by Lafaurie et al. can be explained by the numerical implementations in *interFoam*, which are most likely different than the implementation of Lafaurie et al. For example, Scardovelli and Zaleski [33] report that the connected marker method of Tryggvason and coworkers, from unpublished lecture notes, obtained a slope of $K = 10^{-5}$. This is not surprising as the connected marker method is one of the most precise and time consuming method for simulating multiphase flows. We therefore expect the parasitic currents to be small. The code used by Lafaurie et al. and the *interFoam* solver are both based on the VOF method and include a CSF approach for modelling surface tension. However, Lafaurie et al. produced their results twenty years ago, and it is reasonable to assume the techniques and algorithms in *interFoam* have been improved during this time. This is a possible explanation for why results of *interFoam* have less parasitic currents than Lafaurie et al.

5.4.2 Correlation of Harvie et al.

As explained in Chapter 3, Section 3.2, Harvie et al. show that the maximum parasitic current magnitude obey the following correlation:

$$U_p = \min(a_T U_T, a_A U A, a_V U_V)\,,$$

where the expression of U_T, U_A, and U_V are shown in Chapter 3, Section 3.2 and a_T, a_A, and a_V are coefficients depending on the numerical method. The correlation of Harvie et al. has not been compared to simulations with *interFoam* because of a lack of time. However, some data points are shown in Figure 5.10.

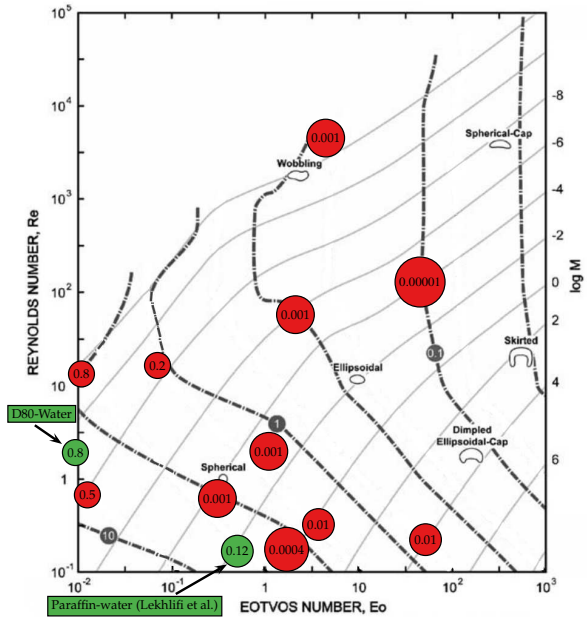


Figure 5.10: Parasitic current magnitudes plotted as a function of the Eötvös and Reynolds number for fluid particles moving at terminal velocity. M is the Morton number. The phase chart is based on data from Clift et al. [9] and Harvie et al. [16]. The dashed-dot grey contours show predicted parasitic current magnitudes for a water drop falling in various silicone oils. Intermediate contour levels are 0.0333, 0.333, etc. Red dots show parasitic currents with *interFoam*. Green dots indicate the simulation of Exxsol D80-water and paraffin-water systems. The parasitic current magnitudes are all normalised by the simulated terminal velocity.

Harvie et al. suggest that a VOF code with the CSF approach can only simulate within the spherical cap or skirted regime, if one wishes to limit parasitic currents to 0.1 times the terminal velocity magnitude. However, the conclusion of the quick study in *interFoam*, shown in Figure 5.10, suggests that parasitic currents

are generally smaller than the parasitic currents magnitude obtained by Harvie et al. The general trend obtained by Harvie et al. seem to be kept with *interFoam*: surface tension dominated flow yields the most parasitic currents. Results with *interFoam* yield parasitic currents some order of magnitude lower than Harvie et al.'s results. This may be attributed to a difference in the coding and numerics of the two solvers because Harvie et al. used the Rudman implementation of the CSF technique [16]. Additionally, their results were published in 2006, so a general improvement of the coding in *interFoam* is a safe hypothesis.

5.5 Origin of velocity oscillations

Some results presented earlier, for example Figure 5.2 or Figure 5.11, show that the velocity magnitude of the drop oscillates with a given frequency. This is due to the surface tension. Figure 5.12 shows how the drop oscillates from left to right.

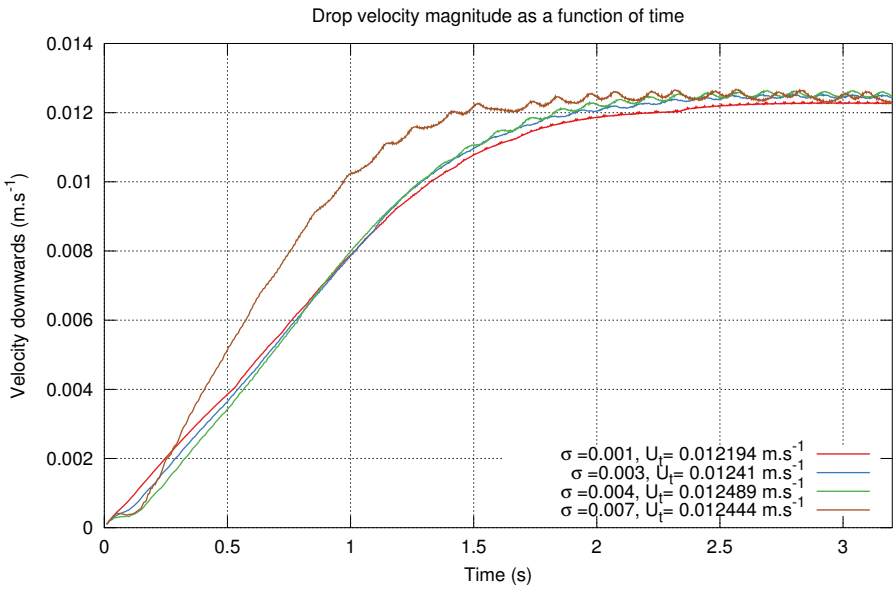


Figure 5.11: Plot showing velocity oscillations for different surface tensions. Drop diameter is $d = 500$ µm and other fluid properties are the one of Exxsol D80 oil and water.

Increasing the surface tension results in the drop oscillating sooner, as well as increasing the amplitude of the oscillations. This trend was confirmed by data points ranging from $\sigma = 0.001\ \mathrm{N\,m^{-1}}$ to $\sigma = 0.01\ \mathrm{N\,m^{-1}}$, but only some intermediate data points are shown in Figures 5.11 and 5.12. A high surface tension will generate parasitic currents strong enough to modify the path of the drop, as seen

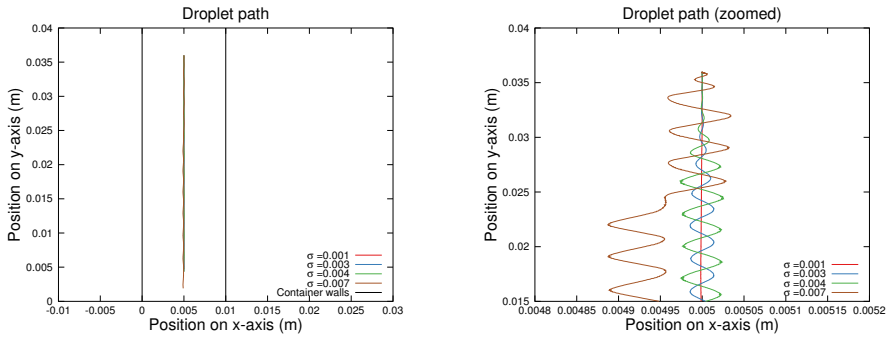


Figure 5.12: Plot showing the drop path for different surface tensions. The graph on the left shows the path of the drop to the scale of the container and is zoomed-in to the right. A too high surface tension generates parasitic currents strong enough to change the path of the drop. Drop diameter is $d = 500$ µm and other fluid properties are the one of Exxsol D80 oil and water.

with the highest surface tension in Figure 5.12. The frequency of the oscillations are the same but no explanation has been found for this phenomenon. The amplitude of the oscillations are higher when increasing the surface tension. This may be attributed to the drop shape because for a lower surface tension small velocity fluctuations are more easily absorbed by the drop. For a high surface tension, the drop is closer to a solid particle and the velocity fluctuations around the drop will affect the drop path instead of the drop shape.

5.6 Contour plot of parasitic currents in OpenFoam

The magnitude of parasitic currents using the *interFoam* solver is investigated for different surface tension values and drop viscosities. The ratio of the parasitic current magnitude and the terminal velocity is calculated to show how much the parasitic currents affects the terminal velocity.

The terminal velocity is found as described in Chapter 4. The parasitic current magnitude is found by switching off the gravity field in the "transportProperties" file.

Simulations have been run for surface tensions and viscosities equal to:

- $\sigma = 0.001$ N m^{-1}
- $\sigma = 0.01$ N m^{-1}
- $\sigma = 0.016$ N m^{-1}
- $\sigma = 0.022$ N m^{-1}
- $\sigma = 0.0344$ N m^{-1}
- $\mu_d = 0.001$ Pa s
- $\mu_d = 0.025$ Pa s
- $\mu_d = 0.05$ Pa s
- $\mu_d = 0.1$ Pa s.

All combinations of these parameters yield 20 data points which are used to draw the contour plot in Figure 5.13, for a drop with diameter $d = 500$ µm. A similar contour plot in Figure 5.14 is done for a drop with diameter $d = 1$ mm.

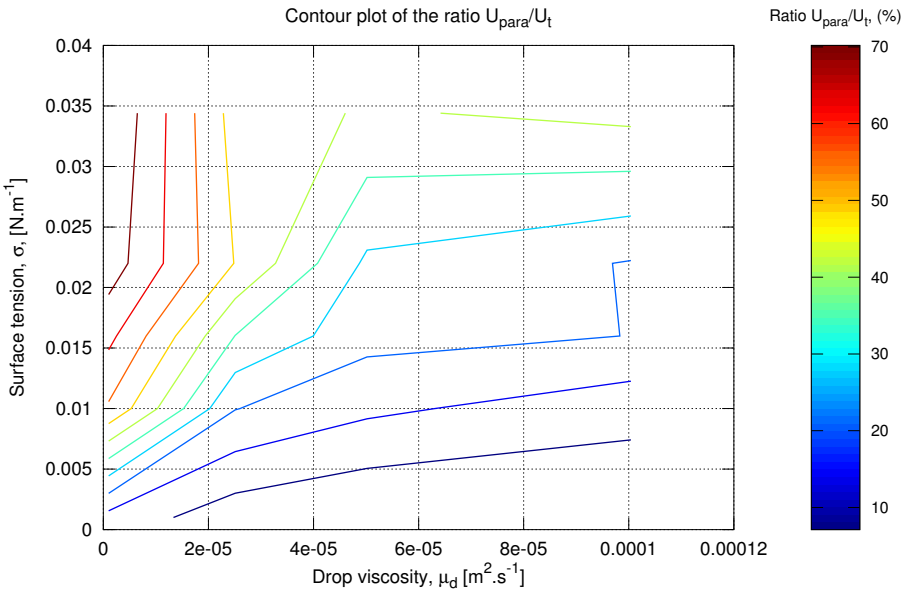


Figure 5.13: Contour plot of the error caused by parasitic currents, depending on surface tension and viscosity for a $d = 500$ µm drop. The ratio of magnitude of parasitic currents to terminal velocity is plotted for different surface tensions and drop viscosities. U_{para} and U_t denotes the parasitic current magnitude and terminal velocity, respectively.

Figure 5.13 and 5.14 show that simulations with a high surface tension yield unusable results. However, increasing the viscosity of the drop lowers error due to parasitic currents. This is in agreement with results of Lafaurie et al. and Harvie et al.

The comparison of Figure 5.13 and Figure 5.14 show that parasitic currents are generally higher for the drop with diameter $d = 500$ µm. One can reasonably assume this trend will be kept for even smaller drops. Simulating a drop with diameter $d = 200$ µm was often impossible. The drop exited the domain in a very short time, suggesting the parasitic currents were very strong. There was no time for thorough testing with smaller or larger drops.

Note: When parasitic currents were strong, the terminal velocity of the drop was difficult to find because its path was seldom straight and the velocity magnitude fluctuated a lot. Some simulations had parasitic currents large enough to counterbalance gravity, thus making the drop go in a random motion. These simulations were not taken into account, but simulations in Figure 5.13 and 5.14 with

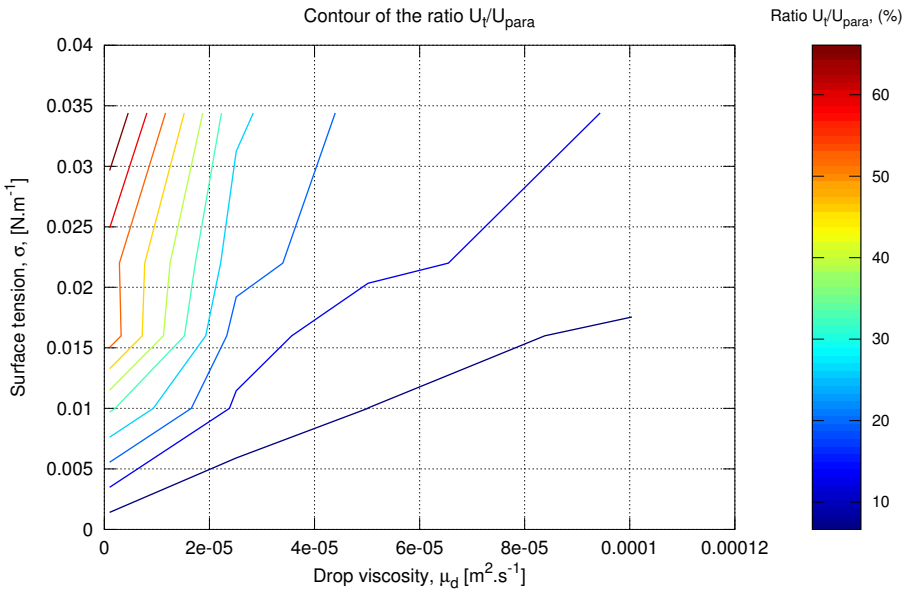


Figure 5.14: Contour plot of the error caused by parasitic currents, depending on surface tension and viscosity for a $d = 1$ mm drop. The ratio of magnitude of parasitic currents to terminal velocity is plotted for different surface tensions and drop viscosities. U_{para} and U_t denotes the parasitic current magnitude and terminal velocity, respectively.

high error are at the borderline of these effects. There is therefore a high uncertainty in the terminal velocity of the drop with gravity. The positive side is that it does not really matter: the parasitic currents are too strong to yield interesting results anyway.

5.7 Influence of drop diameter

Experiments carried out at NTNU on the terminal velocity of water drops in Exxsol D80 oil show good agreement with the Intermediate law, as shown in Figure 5.15.

Simulations with different drop diameter will now be presented. For drops with small diameters, parasitic currents were very strong and caused the drop to go in a random motion. These simulations were not kept.

Figure 5.16 shows the effect of changing surface tension. Simulations with high surface tension yield bad results due to parasitic currents: the terminal velocity is not changing with the diameter of the drop. Drops with lower surface tension seem to converge towards the Intermediate law. There is no certitude, as the effect of parasitic current on drops smaller than 500 µm was not studied due to a lack of time.

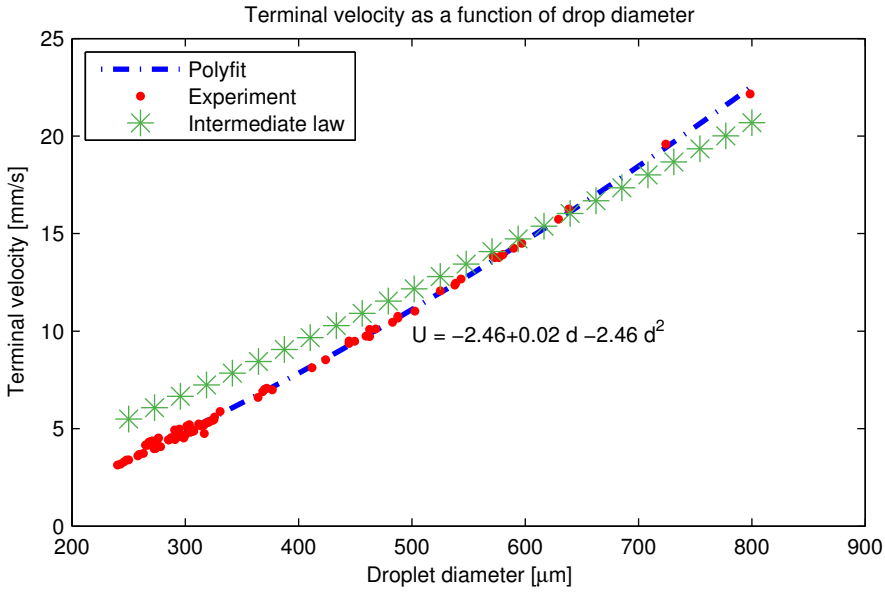


Figure 5.15: Experimental data on terminal velocity of water drops in Exxsol D80 oil obtained at NTNU. Graph is from *Study of droplet-interface dynamics in oil-water separators* by J. Bjerknes [7]. Experimental points follow the Intermediate law.

Figure 5.17 shows the effect of changing the drop viscosity on the terminal velocity as a function of drop diameter. Looking at the contour plot in Figure 5.13 and 5.14, increasing viscosities reduces the effect of parasitic currents. Increasing the viscosity makes the plot converge towards the Intermediate law. This is reasonable as a viscous drop is closer to predictions from a solid sphere model.

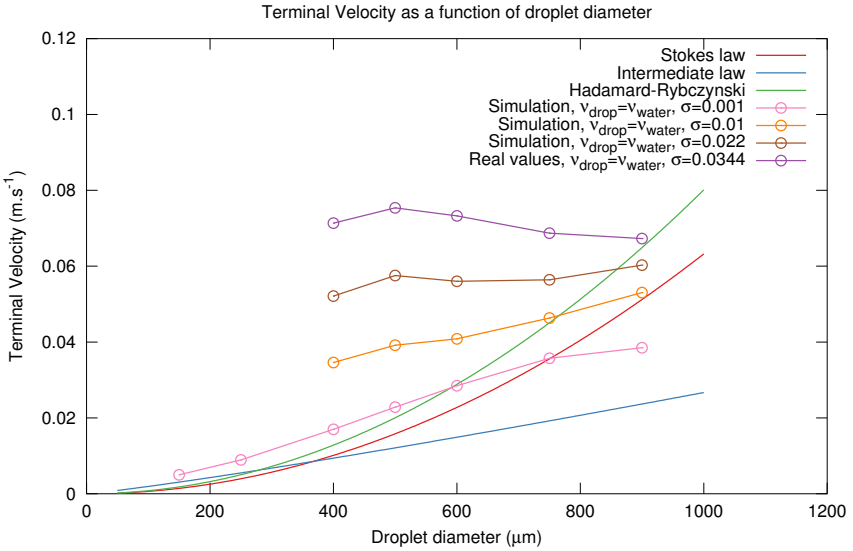


Figure 5.16: The terminal velocity is plotted as a function of drop diameter. Surface tension is varied. Drops with small diameter presented high parasitic currents hindering the fall of the drop and are therefore not included.

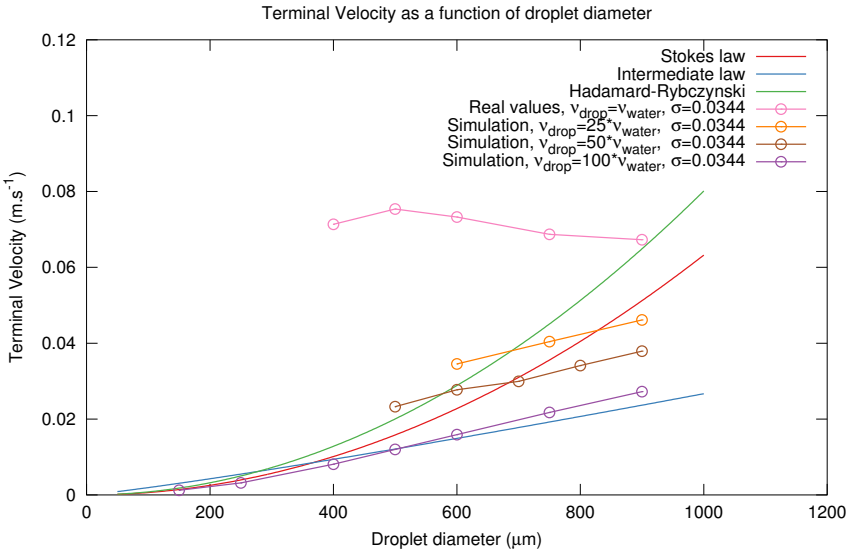


Figure 5.17: The terminal velocity is plotted as a function of drop diameter. The drop viscosity is increased. Drops with small diameter presented high parasitic currents hindering the fall of the drop and are therefore not included.

CHAPTER 6

Conclusion

The goal of this work was to evaluate the possibility of using OpenFoam and a VOF approach to simulate the terminal velocities of small water drops in Exxsol D80 oil.

The analysis of parasitic currents in OpenFoam yielded interesting results. Results from the literature were replicated with relatively good agreement. This analysis showed that the VOF method in the *interFoam* solver of OpenFoam can not be used to simulate the fall of drops using physical properties of water and Exxsol D80 oil and small diameters because parasitic currents are too strong, up to 70% of the terminal velocity. When increasing the viscosity of the drop and decreasing the surface tension, parasitic currents decreased to 10% of the terminal velocity. Additionally, results showed good agreement with the theoretical predictions of the Intermediate law and experimental results obtained at NTNU.

Several points should be further studied. In the attempt to simulate water drops in Exxsol D80 oil without strong parasitic currents, in the diameter range of 100 µm to 1000 µm, several options exist. First of all, increasing the drop viscosity in order to simulate a very viscous sphere showed good agreement with the Intermediate law. Experimental results obtained at NTNU also show that the water drops in the diameter range considered and in Exxsol D80 oil follow the Intermediate law. Experimental results may be replicated by the VOF method in OpenFoam, but further testing should be done, by increasing the drop viscosity even more and studying parasitic currents for even smaller drops for example.

Lastly, using more advanced methods is an option to replicate results from the water-Exxsol D80 experimental setup of NTNU. The Level-Set method or the Connected Marker Method are more complex and advanced algorithms. These methods require higher computing power, but might yield better results when simulating drops in the range considered. Another idea is to study the fall of solid particles, by using a single phase solver and letting a solid sphere with the correct density fall through the ambient fluid. This can be an easy solution to study the fall of drop clusters.

Several other aspects could have been studied if given more time. Fresh ex-

perimental results obtained at NTNU during the Spring 2013 show that the temperature influences the terminal velocity of water drop in crude oil. Including the energy equation in the simulations would be interesting to study. This means find another solver in OpenFoam which includes the thermal aspect of the problem.

It would also be desirable to run simulations on high performance clusters, to check if increasing the mesh resolution yields results closer to the Hadamard-Rybczynski law. We suspect the coarse grid did not allow internal circulation in the drop to show. However, experimental results points to drops following the terminal velocity of rigid spheres, thus without internal circulation. Anyhow, increasing the mesh might show some surprising results. A 2D case is able to replicate theoretical results, but simulating a liquid drop in 3D would also be interesting to try out.

Finally, an extensive quantitative analysis of parasitic currents in OpenFoam for several ranges of viscosity and surface tension is highly desirable. We did not have time to study this extensively, but it would be interesting to map exactly the ranges for which OpenFoam yields good results in drop simulations. The whole OpenFoam community is interested in a quantitative analysis of parasitic currents, not only for drops, but for all kinds of multiphase flows. Such a study would be useful for evaluating possible source of errors in multiple multiphase flows simulations. There are numerous experimental results in other drop ranges and replicating such results with OpenFoam would also be interesting.

Bibliography

[1] Openfoam internet community. http://www.cfd-online.com/Forums/openfoam/.

[2] Openfoam programmer's guide. http://www.foamcfd.org/Nabla/guides/ProgrammersGuide.html, 2007.

[3] Information on the swak4foam toolbox. http://openfoamwiki.net/index.php/Contrib/swak4Foam, 2012.

[4] Openfoam user guide. www.openfoam.org/docs, 2012.

[5] Exxsol D80 product datasheet. http://exxonmobilchemical.ides.com/en-US/ds129003/Exxsol%E2%84%A2%20D80.aspx?I=22455&U=0, 2013.

[6] K. Bäumler, M. Wegener, A.R. Paschedag, and E. Bansch. Drop rise velocities and fluid dynamic behavior in standard test systems for liquid/liquid extraction-experimental and numerical investigations. *Chemical Engineering Science*, 2010.

[7] J.V. Bjerknes. Study of droplet-interface dynamics in oil-water separators, 2012. Project thesis.

[8] J.U. Brackbill, D.B. Kother, and C. Zemach. A continuum method for modelling surface tension. *Journal of Comparative Physiology*, 1991.

[9] R. Clift, J.R. Grace, and M.E. Weber. *Bubbles, Drops and Particles*. Dover Publications, 2005.

[10] R. Cusak. Rethinking your liquid-liquid separations. *Hydrocarbon processing*, 2009.

[11] S. Márquez Damián. Description and utilization of interFoam multiphase solver.

[12] T.J. Danielson and ConocoPhillips Co. Sand transport modelling in multiphase pipelines. *Offshore Technology Conference, Houston, Texas, U.S.A*, 2007.

[13] K.B Deshpande and W.B. Zimmerman. Simulation of interfacial mass transfer by droplet dynamics using the level set method. *Chemical Engineering Science*, 2006.

[14] R.T. Eiswirth, H.-J. Bart, T. Atmakidis, and E.Y. Kenig. Experimental and numerical investigation of a free rising droplet. *Chemical Engineering and Processing: Process Intensification*, 2011.

[15] J.R. Grace, T. Wairegi, and T.H. Nguyen. Shapes and velocities of single drops and bubbles moving freely through immiscible liquids. *Trans. Inst. Chem. Eng.*, 1976.

[16] D.J.E. Harvie, M.R. Davidson, and M. Rudman. An analysis of parasitic current generation in Volume of Fluid simulations. *Applied Mathematical Modelling*, 2006.

[17] J. Hass, M.D. Weir, and G.B. Thomas Jr. *Calculus.* Pearson Custom Publication, 2007.

[18] C.W. Hirt and B.D. Nichols. Volume of fluid (VOF) method for the dynamics of free boundaries, 1979.

[19] S. Hu and R.C. Kintner. The fall of single liquid drops through water. *AiChE journal*, 1955.

[20] A.J. Klee and R.E. Treybal. Rate of rise or fall of liquid drops. *AIChE Journal*, 1956.

[21] J. Klostermann, K. Schaake, and R. Schwarze. Numerical simulation of a single rising bubble by VOF with surface compression. *International Journal for Numerical methods in Fluids*, 2012.

[22] P.M. Krishna, D. Venkateswarlu, and G.S.R. Narasimhamurty. Fall of liquid drops in water, peak velocities and maximum drop size. *J. Chem. Eng. Data.*, 1959.

[23] A. Kumar. *Droplet behaviour in liquid/liquid extraction.* PhD thesis, ETH Zurich, 1983.

[24] B. Lafaurie, C. Nardone, R. Scardovelli, S. Zaleski, and G. Zanetti. Modelling merging and fragmentation in multiphase flows with SURFER. *Journal of Computational Physics*, 1993.

[25] A. Lekhlifi, M. Antoni, and J. Ouazzani. Numerical simulation of the unsteady hydrodynamics of a water droplet in paraffin oil. *Colloids and Surface: Physicochemical and Engineering Aspects*, 2009.

[26] W. Licht and G.S.R. Narasimhamurty. Rate of fall of single liquid droplets. *AIChE Journal*, 1955.

[27] S. Mokhatab, W.A. Poe, and J.G.Speight. *Handbook of Natural gas Transmission and processing.* Gulf Professional Publishing, 22006.

[28] M. Mousavichoubeh, M. Ghadiri, and M. Shariaty-Niassar. Electrocoalescence of an aqueous droplet at an oil-water interface. 2011.

[29] B. Müller. Introduction to computational fluid dynamics, 2012. Lecture notes for TEP4165-Computational Heat and Fluid flow at NTNU.

[30] S. Osher and R. Fedkiw. Level set methods: an overview and some recent results. *Journal of Computational Physics*, 2001.

[31] J.H. Perry. *Chemical Engineers Handbook.* McGraw-Hill, 3rd edition, 1950.

[32] N. Samkhaniani, A. Ajami, M.H. Kayhani, and A.S. Dari. Direct numerical simulation of single bubble rising in viscous stagnant liquid. *International Conference on Merchanical, Automobile and Robotics Engineering (ICMAR'2012)*, 2012.

[33] R. Scardovelli and S. Zaleski. Direct numerical simulation of free-surface and interfacial flow. *Annual Review of Fluid Mechanics*, 1999.

[34] M. Stewart and K. Arnold. *Gas-Liquid and Liquid-Liquid separators.* 2008.

[35] M. Sveier. Study of the droplet-interface dynamics related to liquid-liquid separators. Master's thesis, NTNU, June 2012.

[36] S. Tanguy, T. Menard, A. Berlemont, J.L. Estivalezes, and F. Couderc. Level set method for interface tracking: development and applications. *La houille blanche*, 2006.

[37] C.W. Ueberhuber. *Numerical Computation-Methods, Software, and Analysis.* Springer, 1997.

[38] H.K. Versteeg and W. Malalasekera. *An Introduction to Computational Fluid Dynamics. The Finite Volume Method.* Pearson, Harlow, 2007.

[39] F.M. White. *Fluid Mechanics.* McGraw-Hill Education, 5^{th} edition, 2003.

APPENDIX A

OpenFoam code and Octave functions

```
downAverage
    {   type swakExpression;
        valueType internalField;
        variables (
             "downDirection=vector(0,-1,0);"
             "thres=0.5;"//which cells to keep regarding phase
                 fraction
             "liquidVol=sum(alpha1>thres ? vol() : 0);" //calculates
                 the volume of the drop
             "downVel=alpha1>thres ? (U & downDirection) : 0;" //a & b
                 :inner vector product. Keep the downwards component
                 of U.);
        expression "downVel*vol()/liquidVol";//vol():vol of the cell
        accumulations (sum);
           sum //weighted average downward velocity
        verbose true;}
```

Code A.1: Extraction of droplet velocity with *swak4Foam*. Velocity magnitude for cells within the drop are added and then divided by the area of the drop.

```
createInterface
    {   type createSampledSurface;
        outputControl timeStep;
        outputInterval 1;
        surfaceName interface;
        surface {
        type isoSurface;
        isoField alpha1;
        isoValue 0.5; //interface of the drop, where alpha1=0.5
        interpolate true;}
    }

xDiameter
    {   type swakExpression;
        valueType surface;
        surfaceName interface;
        verbose true;
        expression "pos().x";//find the x coordinates of the cells
            with alpha1=0.5
        accumulations (
            min //writes down min and max values
            max);
    }

yDiameter
    {   type swakExpression;
        valueType surface;
        surfaceName interface;
        verbose true;
        expression "pos().y"; //find the y coordinates of the cells
            with alpha1=0.5
        accumulations (
            min //writes down min and max values
            max);
    }
```

Code A.2: Extraction of droplet diameter with *swak4Foam*. Maximum and minimum coordinates in the x and y directions are saved.

```
function averageUt=averageTermVel(time,data) #time vector and velocity
     vector as input
Npoints=size(data)(1);#number of data points
tolerance=0.001;#tolerance level for slope of curve
polyCoeff=polyfit(time,data,4);#fits the curve with a polynome of
    order 4
fittedUy=polyCoeff(1,1)*time.^4+polyCoeff(1,2)*time.^3+polyCoeff(1,3)*
    time.^2+polyCoeff(1,4)*time+polyCoeff(1,5); #polyfitted data set,
    smoother
slopeVector=1./(gradient(time,fittedUy));
newUy=zeros(Npoints,1);

for i=1:Npoints
        if abs(slopeVector(i)) < tolerance #keeping the points that
            have stabilized
                newUy(i,1)=data(i,1);
        endif
endfor

averageUt=mean(nonzeros(newUy));

endfunction
```

Code A.3: Octave function for calculating the terminal velocity. A polynome is fitted to the curve. When the slope of the fitted curve is close to zero, terminal velocity is reached.

```
function Ut=calculateUt(Cd)
Ut=sqrt((4*9.81*0.002*140/(3*860*Cd)));#gravity=9.81m/s2,diameter=2cm,
    density difference=140kg/m3
endfunction

function Cd=calculateCd(Re)
if (Re<0.1)
Cd=24/Re;#Stokes law
elseif (Re>0.1 & Re<1000)
Cd=(24/Re)*(1+0.14*Re^0.7);#Intermediate law
elseif (Re>1000)
Cd=0.445;#Newtons law
endif
endfunction

function Re=calculateRe(Ut)
Re=860*0.002*Ut/0.099;#Density=860kg/m3,diameter=2cm,dynamic viscosity
    =0.099 Pa.s
endfunction
```

Code A.4: Octave function to find the theoretical terminal velocity by iterations.